HCFa
— 禾川科技 —

Q Series Medium-Sized PLC

Motion Control Manual

ATC

Catalogue No.ATC/MQXM2210

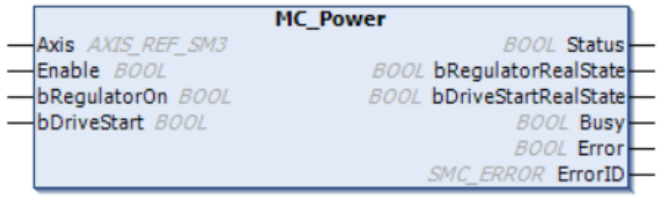
April 29, 2022

Contents

MC_Power (FB)

Enable the axis.

Name	MC_Power (axis enable)		
Supported modes	CSP	CSV	CST
Graphical performance		ST performance _	




```
MC_ Power (
Axis:=,
Enable:=,
bRegulatorOn :=,
bDriveStart :=,
Status= >,
bRegulatorRealState = >,
bDriveStartRealState = >,
Busy = >,
Error = >,
ErrorID => );
```

Variables

(1) Input and output variables

Input and output variable	Name	Type of data	Content
Axis	Axis	AXIS_REF_SM3	Specify the axis, which is an instance of AXIS_REF_SM3

(2) Input variables

Input variable	Name	Type of data	Effective range	Initial value	Content
Enable	Effective	BOOL	TRUE, FALSE	FALSE	TRUE: Enable function block FALSE: Disable the function block
bRegulatorOn	Enable	BOOL	TRUE, FALSE	FALSE	TRUE: Turn on axis enable FALSE : Turn off the axis enable
bDriveStart	Drive on	BOOL	TRUE, FALSE	FALSE	TRUE: Allow axis movement FALSE : Prohibit axis movement

(3) Output variables

Output variable	Name	Type of data	Effective range	Content
Status	State	BOOL	TRUE, FALSE	TRUE: The axis can execute motion control function blocks
bRegulatorRealState	Axis enable signal status	BOOL	TRUE, FALSE	TRUE: Axis enable is valid
bDriveStartRealState	Drive on state	BOOL	TRUE, FALSE	TRUE: Allow axis movement

Busy	Executing	BOOL	TRUE, FALSE	TRUE: Function block is running
Error	Error	BOOL	TRUE, FALSE	TRUE: The function block generates an exception and has stopped execution
ErrorID	Error code	SMC_ERROR	0	The fault codes when function block is abnormal

(4) Conversion timing of output variables

Variable	When it becomes TRUE	When it becomes FALSE
Status	✧ Axes can execute motion control function blocks	✧ Axis is not movable ✧ Error becomes TRUE
bRegulatorRealState	✧ Axis enable is effective	✧ Axis enable failure
bDriveStartRealState	✧ Allow axis movement	✧ The axis is not allowed to run
Busy	✧ The rising edge of Enable	✧ When Enable becomes FALSE ✧ When Error becomes TRUE
Error	✧ When a fault occurs during the execution of a function block	✧ When the exception is resolved

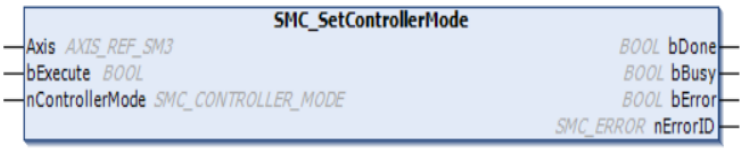
Key points

- Enable is set to TRUE, MC_Power enters the execution state, and other input pins are processed by MC_Power at this time.
- If Enable is set to FALSE, MC _Power will no longer execute any function block whilst modifying the other input pins of MC _Power will have no effect.
- The bRegulatorOn (enable) pin controls the axis enable. When bRegulatorOn is TRUE, MC_Power will try to enable the axis. After the axis is successfully enabled, the bRegulatorRealState pin of MC_Power will output TRUE. When bRegulatorOn is FALSE, MC_Power will disconnect the axis enable, and bRegulatorRealState pin output is FALSE.
- The bDriveStart (drive on) pin allows the axis to move, when bDriveStart is TRUE, MC_Power allows the axis to move, bDriveStartRealState pin output is TRUE; when bDriveStart is FALSE, MC_Power prohibits the axis to move, bDriveStartRealState pin output is FALSE.
- When the axis is moving, turn off the bRegulatorOn or bDriveStart of MC _Power. At this time, the function block CommmandAborted being executed becomes TRUE, the axis is switched to the non-movable state, and all motion control is stopped.
- When bRegulatorRealState is TRUE, bDriveStartRealState is TRUE, and the axis has no faults, the Status pin will output TRUE, the axis state will be converted to StandStill, and other motion control function blocks can be executed.

➢ ※ Note: When disconnecting the axis enable, please do not turn off the bEnable and bRegulatorOn of MC_Power at the same time. This operation may cause that the axis cannot be disabled. It is recommended not to turn off the Enable pin of MC_Power.

SMC_SetControllerMode (FB)

Set the control mode of the drive.

Name	SMC_ SetControllerMode (change control mode)		
Supported modes	CSP	CSV	CST
Graphical performance		ST performance _	
		SMC_ SetControllerMode (Axis:=, bExecute :=, nControllerMode :=, bDone = >, bBusy = >, bError = >, nErrorID =>);	

Variables

(1) Input and output variables

Input and output variable	Name	Type of data	Content
Axis	Axis	AXIS_REF_SM3	Specify the axis, which is an instance of AXIS_REF_SM3

(2) Input variables

Input variable	Name	Type of data	Effective range	Initial value	Content
bExecute	Start up	BOOL	TRUE, FALSE	FALSE	Enter a function block to start executing when it rises
nControllerMode	Control mode	SMC _CONTROLLER_MODE		SMC _Position	Select the control mode, refer to SMC_Controller_Mode

(3) Output variables

Output variable	Name	Type of data	Effective range	Content
bDone	Mode setting completed	BOOL	TRUE, FALSE	TRUE: The execution of the function block is complete
bBusy	Function block is executing	BOOL	TRUE, FALSE	TRUE: The function block is running
bError	Error	BOOL	TRUE, FALSE	TRUE: The function block generates an exception and has stopped execution
nErrorID	Error Code	SMC_ERROR	0	When an exception occurs, the error code is output

(4) Conversion timing of output variables

Variable	When it becomes TRUE	When it becomes FALSE
bDone	✧ When finished	✧ bExecute is TRUE at the same time as the FALSE of bExecute

		✧ bExecute is FALSE, after 1 cycle
bBusy	✧ The rising edge of bExecute	✧ bDone is TRUE ✧ When bError is TRUE ✧ bCommandAborted becomes TRUE
bError	✧ When a fault occurs during the execution of a function block	✧ bExecute is TRUE at the same time as the FALSE of bExecute ✧ bExecute is FALSE, after 1 cycle

Key points

- The rising edge of bExecute is to start the function block. The rising edge can be triggered again during the execution of the function block. The input parameters of the function block will be reloaded each time the rising edge is executed, and the function block will be executed again.
- The control mode of the drive can be set through nControllerMode. For the specific mode conversion, refer to SMC_Controller_Mode.
- Axis must be a physical axis. This function block modifies the axis byControllerMode (command value of the control mode) parameter, and will wait for the byRealControllerMode (actual command of the control mode) parameter to return.
- The specified control mode must be supported. For details, please refer to the instruction manual of the corresponding drive of the axis.
- The communication data of the corresponding mode needs to be configured first. For example, in the torque mode, communication parameters such as target torque need to be configured.
- When the function block is called, the axis cannot be in ErrrorStop, Homing or Stopping state, otherwise an error (SMC_SCM_AXIS_IN_WRONG_STATE) will occur.
- After 1000 cycles of execution of the function block, if it still does not become the set control mode, the function block will be interrupted and an error will be returned.
- When the function block output is bDone, it is necessary to immediately use the function block supported by the corresponding mode to take over the motion control of the axis, otherwise the motion of the axis will be in an unsupervised state and cause a fault.

MC_SetPosition (FB)

To modify the function block position and feedback position of the axis to the position data input by the function block.

Name	MC _ Set Position _		
Supported modes	CSP	CSV	CST
Graphical performance		ST performance _	
		MC_ SetPosition (Axis:=, Execute:=, Position:=, Mode:=, Done= >, Busy = >, Error = >, ErrorID =>);	

Variables

(1) Input and output variables

Input and output variable	Name	Type of data	Content
Axis	Axis	AXIS_REF_SM3	Specify the axis, which is an instance of AXIS_REF_SM3

(2) Input variables

Input variable	Name	Type of data	Effective range	Initial value	Content
Execute	Start up	BOOL	TRUE, FALSE	FALSE	TRUE: Enable function block
Position	Target location	LREAL	Negative number, positive number, "0"	0	Specify the set target position, the unit is [command unit]
Mode	Mode selection	BOOL	TRUE, FALSE	FALSE	TRUE: Relative FALSE: Absolute

(3) Output variable

Output variable	Name	Type of data	Effective range	Content
Done	Finish	BOOL	TRUE, FALSE	TRUE: The execution of the function block is complete
Busy	Function block is executing	BOOL	TRUE, FALSE	TRUE: The function block is running
Error	Error	BOOL	TRUE, FALSE	TRUE: The function block generates an exception and has stopped execution
ErrorID	Error code	SMC_ERROR	0	When an exception occurs, an error code is output

(4) Conversion timing of output variables

variable	When it becomes TRUE	When it becomes FALSE
Done	✧ When the position shift is completed	✧ When Execute is TRUE, at the same time as FALSE of Execute ✧ Execute is FALSE, after 1 cycle
Busy	✧ Rising edge of Execute	✧ When Done is TRUE ✧ Error is TRUE
Error	✧ When a fault occurs during the execution of a function block	✧ When the exception is resolved

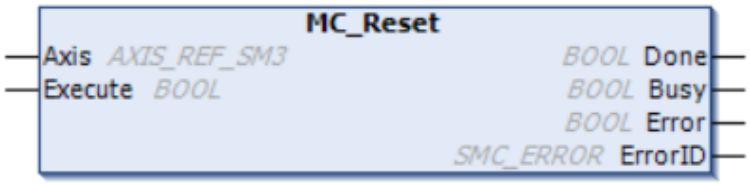
Key points

- The rising edge of Execute is to start the function block. The rising edge can be triggered again during the execution of the function block. The input parameters of the function block will be reloaded each time the rising edge is executed, and the function block will be executed again.
- This function block does not modify the actual encoder position, but only generates a coordinate offset to the current coordinate system.
- Mode pin is TRUE, it is the relative mode, and MC_SetPosition will add the value of the Position pin to the current axis as the position after the coordinate system offset.
- Mode pin is FALSE, it is in absolute mode, and M_SetPosition will directly use the value of the Position pin as the offset position.

➢ ※ Note: The offset generated by this function block will be lost after the controller is powered off, so it cannot replace the MC_Home function block.

MC_Reset (FB)

Reset axis status, to change axis from ErrrorStop to StandStill status, or reset drive fault.

Name	MC_Reset (axis reset)		
Supported modes	CSP	CSV	CST
Graphical performance		ST performance _	
		MC_ Reset (Axis:=, Execute:=, Done= >, Busy = >, Error = >, ErrorID =>);	

Variables

(1) Input and output variables

Input and output variable	Name	Type of data	Content
Axis	Axis	AXIS_REF_SM3	Specify the axis, which is an instance of AXIS_REF_SM3

(2) Input variable

Input variable	Name	Type of data	Effective range	Initial value	Content
Execute	Start up	BOOL	TRUE, FALSE	FALSE	TRUE: Enable function block

(3) Output variables

Output variable	Name	Type of data	Effective range	Content
Done	Finish	BOOL	TRUE, FALSE	TRUE: The execution of the function block is complete
Busy	Function block is executing	BOOL	TRUE, FALSE	TRUE: The function block is running
Error	Error	BOOL	TRUE, FALSE	TRUE: The function block generates an exception and has stopped execution
ErrorID	Error code	SMC_ERROR	0	When an exception occurs, an error code is output

(4) Conversion timing of output variables

Variable	When it becomes TRUE	When it becomes FALSE
Done	✧ When finished	✧ When Execute is TRUE, at the same time as FALSE of Execute ✧ Execute is FALSE, after 1 cycle
Busy	✧ Rising edge of Execute	✧ When Done is TRUE ✧ When Error is TRUE ✧ Command Aborted becomes TRUE
Error	✧ When a fault occurs during the execution of a function block	✧ When the exception is resolved

Key points

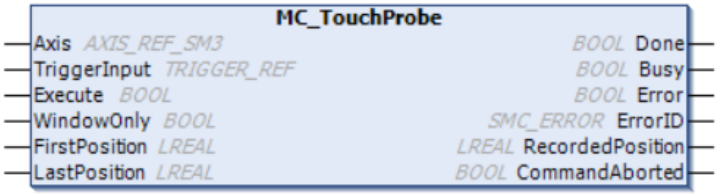
- The rising edge of Execute is to start the function block. The rising edge can be triggered again during the execution of the function block. The input parameters of the function block will be reloaded each time the rising edge is executed, and the function block will be executed again.
- When the axis communication is normal, this function block changes the axis from Errorstop to Standstill, changes the abnormal state of the axis to a normal and operable state, and resets the drive fault state at the same time.
- If the axis uses the MC_Reset function block in other states, an error will be reported, the axis will not report an error, and the current axis state will not be interrupted.

Failure to reset communication-related failures.

➢ ※ Note: Communication-related failures need to re-establish the connection between the master station and the slave station. After the communication is normal, call the SMC3_ReInitDrive function block to reset. This function block can be used for all types of axes.

MC_TouchProbe (FB)

The position of the axis when the signal is triggered by the trigger probe signal.

Name	MC _TouchProbe (Enable external lock)		
Supported modes	CSP	CSV	CST
Graphical performance		ST performance _	
		MC_ TouchProbe (Axis:=, TriggerInput :=, Execute:=, WindowOnly :=, FirstPosition :=, LastPosition :=, Done= >, Busy = >, Error = >, ErrorID = >, RecordedPosition = >, CommandAborted =>);	

Variables

(1) Input and output variables

Input and output variable	Name	Type of data	Content
Axis	Axis	AXIS_REF_SM3	Specify the axis, which is an instance of AXIS_REF_SM3
TriggerInput	Trigger pulse input	TRIGGER_REF	Related to the rising edge signal, refer to TRIGGER_REF

(2) Input variables

Input variable	Name	Type of data	Effective range	Initial value	Content
Execute	Start up	BOOL	TRUE, FALSE	FALSE	TRUE: Enable function block
WindowOnly	Window is valid	BOOL	TRUE, FALSE	FALSE	If TRUE, the trigger signal is only valid within the specified range
FirstPosition	Start position	LREAL	-	0	Specify the start position to receive the trigger signal
LastPosition	End position	LREAL	-	0	Specify the end position of receiving the trigger signal

(3) Output variables

Output variable	Name	Type of data	Effective range	Content
Done	Finish	BOOL	TRUE, FALSE	TRUE: The execution of the function block is complete
Busy	Function block is executing	BOOL	TRUE, FALSE	TRUE: The function block is running
Error	Error	BOOL	TRUE, FALSE	TRUE: The function block generates an exception and has stopped execution
ErrorID	Error code	SMC_ERROR	0	When an exception occurs, an error code is output
RecordedPosition	Record location	LREAL	0	The position recorded when the trigger signal occurs
CommandAborted	Function block execution interrupt	BOOL	TRUE, FALSE	TRUE: The function block is aborted

(4) Conversion timing of output variables

Variable	When it becomes TRUE	When it becomes FALSE
Done	✧ When finished	✧ When Execute is TRUE, at the same time as FALSE of Execute ✧ Execute is FALSE, after 1 cycle
Busy	✧ Rising edge of Execute	✧ When Done is TRUE ✧ When Error is TRUE ✧ CommandAborted becomes TRUE
CommandAborted	✧ Multi-start (interruption) of motion function blocks by other function blocks, when this function block is terminated ✧ When this function block is terminated due to an abnormality ✧ When an exception is occurring, this function block is started ✧ MC _Stop function block is being executed, this function block is	✧ Execute is TRUE, it is at the same time as FALSE of Execute ✧ Execute is FALSE, after 1 cycle

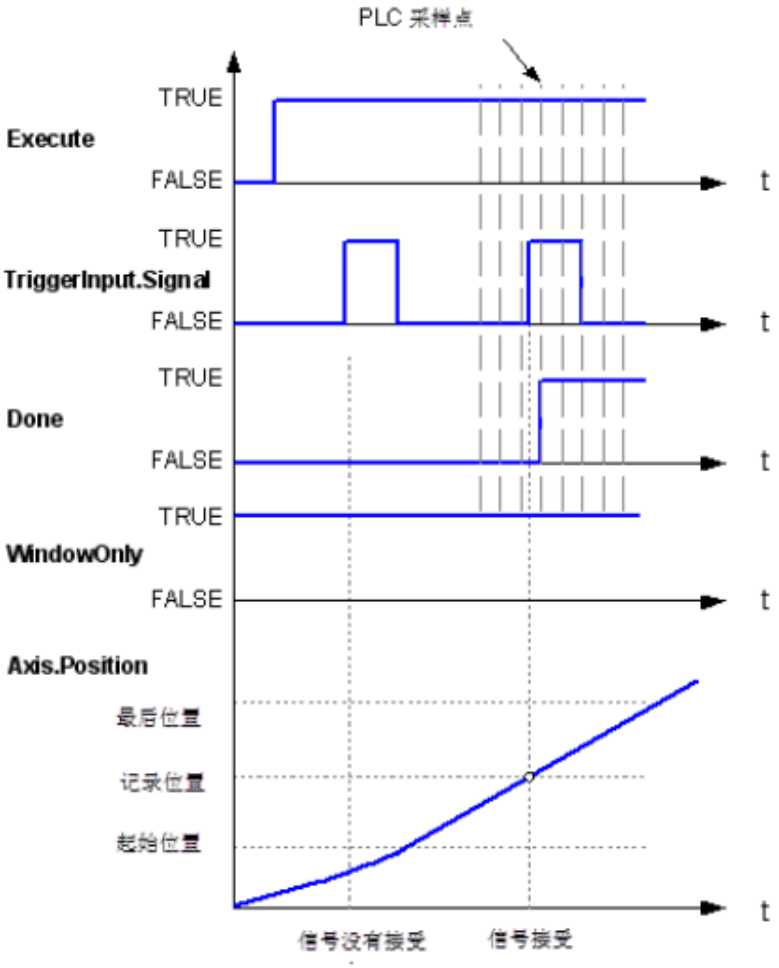
	started	
Error	✧ When a fault occurs during the execution of a function block	✧ When the exception is resolved

Key points

- The rising edge of Execute is to start the function block. The rising edge can be triggered again during the execution of the function block. The input parameters of the function block will be reloaded each time the rising edge is executed, and the function block will be executed again.
- This function block uses TriggerInput (trigger pulse input), WindowOnly (window effective), FirstPosition (start position), LastPosition (end position) as trigger settings. When the signal is triggered, the axis position at the time of trigger will be output to RecordedPosition (recording position).

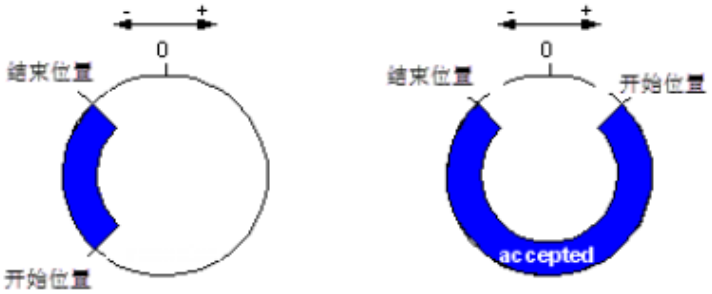
➢ ※ Note: There are two methods to obtain the current position of the axis. It is recommended to use the probe function defined by 16#60B8. Please refer to the servo manual for the specific usage method.

Sequence diagram

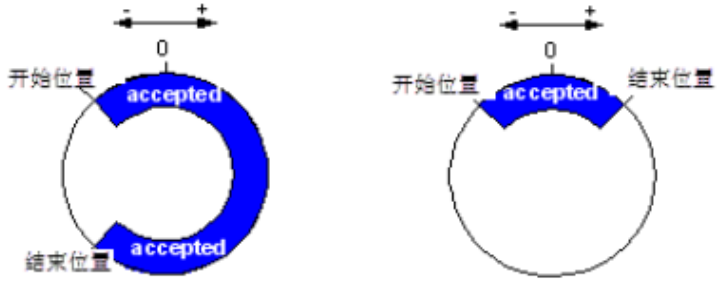


Examples of modal axis

A.　开始位置 < 结束位置



B.　开始位置 > 结束位置



MC_AbortTrigger (FB)

Used with MC_AbortTrigger to break the association with the TriggerInput variables.

Name	MC_AbortTrigger (disable external lock)		
Supported modes	CSP	CSV	CST
Graphical performance		ST performance _	
		MC_ AbortTrigger (Axis:=, TriggerInput :=, Execute:=, Done= >, Busy = >, Error = >, ErrorID =>);	

Variables

(1) Input and output variables

Input and output variable	Name	Type of data	Content
Axis	Axis	AXIS_REF_SM3	Specify the axis, which is an instance of AXIS_REF_SM3
TriggerInput	Trigger pulse input	TRIGGER_REF	Related to the rising edge signal, refer to TRIGGER_REF

(2) Input variables

Input variable	Name	Type of data	Effective range	Initial value	Content
Execute	Start up	BOOL	TRUE, FALSE	FALSE	Start pointing when you enter a rise

(3) Output variables

Output variable	Name	Type of data	Effective range	Content
Done	Finish	BOOL	TRUE, FALSE	TRUE: The execution of the function block is complete
Busy	Function block is executing	BOOL	TRUE, FALSE	TRUE: The function block is running
Error	Error	BOOL	TRUE, FALSE	TRUE: The function block generates an exception and has stopped execution
ErrorID	Error code	SMC_ERROR	0	When an exception occurs, an error code is output

(4) Conversion timing of output variables

Variable	When it becomes TRUE	When it becomes FALSE
Done	✧ When finished	✧ When Execute is TRUE, at the same time as FALSE of Execute ✧ Execute is FALSE, after 1 cycle
Busy	✧ Rising edge of Execute	✧ When Done is TRUE ✧ When Error is TRUE
Error	✧ When a fault occurs during the execution of a function	✧ When the exception is resolved

	block	

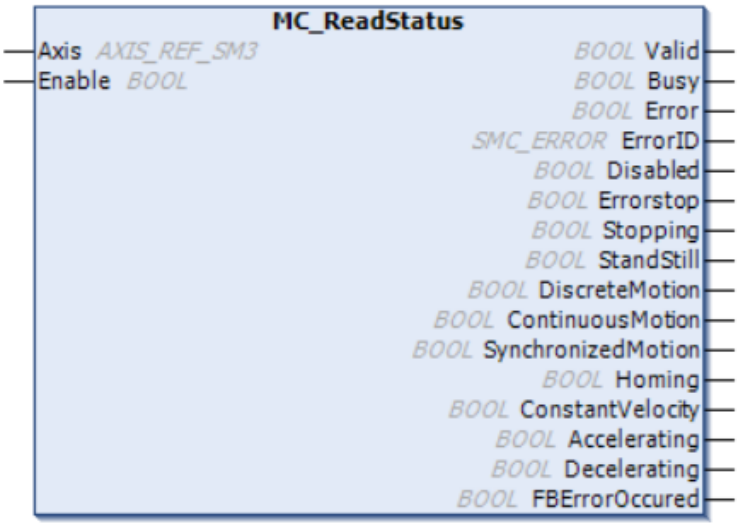
Key points

- The rising edge of Execute is to start the function block. The rising edge can be triggered again during the execution of the function block. The input parameters of the function block will be reloaded each time the rising edge is executed, and the function block will be executed again.
- This function block can be used in conjunction with MC_AbortTrigger by terminating the association between the axis and the TriggerInput (trigger pulse input) variable.

MC_ReadStatus (FB)

Read the current status information of the axis.

Name	MC_ReadStatus (Read axis status)		
Supported modes	CSP	CSV	CST
Graphical performance		ST performance _	




```
MC_ ReadStatus (
Axis:=,
Enable:=,
Valid = >,
Busy = >,
Error = >,
ErrorID = >,
Disabled = >,
Errorstop = >,
Stopping = >,
StandStill = >,
DiscreteMotion = >,
ContinuousMotion = >,
SynchronizedMotion = >,
Homing = >,
ConstantVelocity = >,
Accelerating= >,
Decelerating= >,
FBErrorOccured => );
```

Variables

(1) Input and output variables

Input and output variable	Name	Type of data	Content
Axis	Axis	AXIS_REF_SM3	Specify the axis, which is an instance of AXIS_REF_SM3

(2) Input variables

Input variable	Name	Type of data	Effective range	Initial value	Content
Enable	Effective	BOOL	TRUE, FALSE	FALSE	TRUE: Enable function block FALSE : Close the function block

(3) Output variables

Output variable	Name	Type of data	Effective range	Content
Valid	Axis status data can obtain flags	BOOL	TRUE, FALSE	TRUE: The output is valid
Busy	Function block is executing	BOOL	TRUE, FALSE	TRUE: The function block is running
CommandAborted	Function block execution interrupt	BOOL	TRUE, FALSE	TRUE: The function block is aborted
Error	Error	BOOL	TRUE, FALSE	TRUE: The function block generates an exception and has stopped execution
ErrorID	Error code	SMC_ERROR	0	When an exception occurs, an error code is output
Disabled	Axis not enabled flag	BOOL	TRUE, FALSE	TRUE: The axis is not enabled
Errorstop	Axis error flag	BOOL	TRUE, FALSE	TRUE: The axis is in the wrong operating state
Stopping	Axis stop sign	BOOL	TRUE, FALSE	TRUE: The axis status is Stopping
StandStill	Axis waiting sign	BOOL	TRUE, FALSE	TRUE: The axis state is StandStill
DiscreteMotion	Axis discrete status flag	BOOL	TRUE, FALSE	TRUE: The axis state is DiscreteMotion
ContinuousMotion	Axis continuous state flag	BOOL	TRUE, FALSE	TRUE: The axis state is ContinuousMotion
SynchronizedMotion	Axis synchronization status flag	BOOL	TRUE, FALSE	TRUE: The axis state is SynchronizedMotion
Homing	Axis back to the original state flag	BOOL	TRUE, FALSE	TRUE: The axis state is Homing
ConstantVelocity	Axis uniform speed status flag	BOOL	TRUE, FALSE	TRUE: axis running speed reached
Accelerating	The axis is accelerating sign	BOOL	TRUE, FALSE	TRUE: The axis is accelerating
Decelerating	Axis is decelerating sign	BOOL	TRUE, FALSE	TRUE: The axis is decelerating
FBErrorOccured	Function block error flag	BOOL	TRUE, FALSE	TRUE: A function block error occurred. It has not been cleared by SMC_ClearFBError

(4) Conversion timing of output variables

Variable	When it becomes TRUE	When it becomes FALSE
Valid	✧ The output is valid	✧ Enable is TRUE, it is at the same time as FALSE of Enable ✧ Enable is FALSE, after 1 cycle

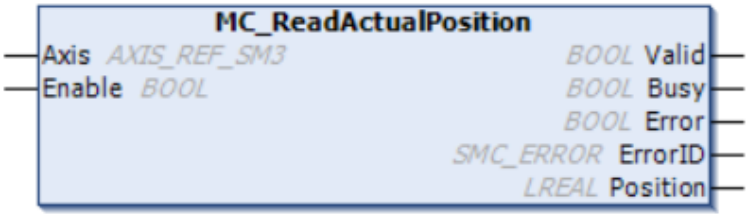
Busy	✧ Enable 's rising edge	✧ When Done is TRUE ✧ When Error is TRUE ✧ Command Aborted becomes TRUE
CommandAborted	✧ Multi-start (interruption) of motion function blocks by other function blocks, when this function block is terminated ✧ When this function block is terminated due to an abnormality ✧ When an exception is occurring, this function block is started ✧ MC _Stop function block is being executed, this function block is started	✧ Enable is TRUE, it is at the same time as FALSE of Enable ✧ Enable is FALSE, after 1 cycle
Error	✧ When a fault occurs during the execution of a function block	✧ When the exception is resolved
Disabled	✧ When the axis is not enabled	✧ When axis is enabled
Errorstop	✧ The axis is in the wrong operating state	✧ The axis is not in the wrong running state
Stopping	✧ When the axis status is Stopping	✧ When the axis status is not S toping
StandStill	✧ When the axis state is StandStill	✧ When the axis state is not StandStill
DiscreteMotion	✧ When the axis state is DiscreteMotion	✧ When the axis state is not D iscreteMotion
ContinuousMotion	✧ When the axis state is ContinuousMotion	✧ When the axis state is not C ontinuousMotion
SynchronizedMotion	✧ When the axis status is SynchronizedMotion	✧ When the axis status is not SynchronizedMotion
Homing	✧ When the axis state is Homing	✧ When the axis status is not Homing
ConstantVelocity	✧ When the axis is moving at a constant speed	✧ axis is not moving at a constant speed
Accelerating	✧ State of the axis when accelerating	✧ The axis is not in the state when accelerating
Decelerating	✧ State of the axis during deceleration	✧ The axis is not in the state during deceleration
FBErrorOccured	✧ When the function block is wrong	✧ When the function block has no error

Key points

- Enable of the function block is TRUE, the axis status information can be updated in real time;
- Enable of the function block is FALSE, the previously updated axis status information will be retained.

MC_ReadActualPosition (FB)

Read the current actual position of the axis.

Name	MC_ReadActualPosition (Read the actual position of the axis)		
Supported modes	CSP	CSV	CST
Graphical performance		ST performance _	
		MC_ ReadActualPosition (Axis:=, Enable:=, Valid = >, Busy = >, Error = >, ErrorID = >, Position=>);	

Variables

(1) Input and output variables

Input and output variable	Name	Type of data	Content
Axis	Axis	AXIS_REF_SM3	Specify the axis, which is an instance of AXIS_REF_SM3

(2) Input variables

Input variable	Name	Type of data	Effective range	Initial value	Content
Enable	Effective	BOOL	TRUE, FALSE	FALSE	TRUE: Enable function block FALSE : Close the function block

(3) Output variables

Output variable	Name	Type of data	Effective range	Content
Valid	Axis status data can obtain flags	BOOL	TRUE, FALSE	TRUE: The output is valid
Busy	Function block is executing	BOOL	TRUE, FALSE	TRUE: The function block is running
Error	Error	BOOL	TRUE, FALSE	TRUE: The function block generates an exception and has stopped execution
ErrorID	Error code	SMC_ERROR	0	When an exception occurs, an error code is output
Position	Position of the axis read	LREAL		Position data of the axis read by the function block

(4) Conversion timing of output variables

Variable	When it becomes TRUE	When it becomes FALSE
Valid	✧ The output is valid	✧ Enable is TRUE, it is at the same time as FALSE of Enable ✧ Enable is FALSE, after 1 cycle
Busy	✧ Enable 's rising edge	✧ When Done is TRUE ✧ When Error is TRUE
Error	✧ When a fault occurs during the execution of a function block	✧ When the exception is resolved

Key points

- This function block can read the actual position of the axis:

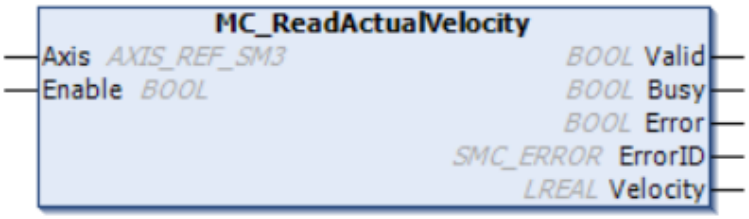
  When Enable is TRUE, the function block updates Position data in real time;

  When Enable is FALSE, the last updated data is retained.

MC_ReadActualVelocity (FB)

Read the current actual speed of the axis.

Name	MC_ReadActualVelocity (Read the actual velocity of the axis)

Supported modes	CSP	CSV	CST
Graphical performance		ST performance _	
		MC_ ReadActualVelocity (Axis:=, Enable:=, Valid = >, Busy = >, Error = >, ErrorID = >, Velocity=>);	

Variables

(1) Input and output variables

Input and output variable	Name	Type of data	Content
Axis	Axis	AXIS_REF_SM3	Specify the axis, which is an instance of AXIS_REF_SM3

Input variables

Input variable	Name	Type of data	Effective range	Initial value	Content
Enable	Effective	BOOL	TRUE, FALSE	FALSE	TRUE: Enable function block FALSE : Close the function block

(2) Output variables

Output variable	Name	Type of data	Effective range	Content
Valid	Axis status data can obtain flags	BOOL	TRUE, FALSE	TRUE: The output is valid
Busy	Function block is executing	BOOL	TRUE, FALSE	TRUE: The function block is running
Error	Error	BOOL	TRUE, FALSE	TRUE: The function block generates an exception and has stopped execution
ErrorID	Error code	SMC_ERROR	0	When an exception occurs, an error code is output
Velocity	Speed of the axis read	LREAL		speed data of the axis read by the function block

(3) Transformation timing of input variables

Variable	When it becomes TRUE	When it becomes FALSE
Valid	✧ The output is valid	✧ Enable is TRUE, it is at the same time as FALSE of Enable ✧ Enable is FALSE, after 1 cycle
Busy	✧ Enable 's rising edge	✧ When Done is TRUE ✧ When Error is TRUE
Error	✧ When a fault occurs during the execution of a function block	✧ When the exception is resolved

Key points

- This function block can read the actual speed in the drive:

  When Enable is TRUE, the function block updates Velocity data in real time;

  When Enable is FALSE, the last updated data is retained.

MC_ReadActualTorque (FB)

Read the current actual torque of the axis.

Name	MC_ReadActualTorque (Read the actual torque of the axis)		
Supported modes	CSP	CSV	CST
Graphical performance		ST performance _	


```
MC_ ReadActualTorque (
Axis:=,
Enable:=,
Valid = >,
Busy = >,
Error = >,
ErrorID = >,
Torque=> );
```

Variables

(1) Input and output variables

Input and output variable	Name	Type of data	Content
Axis	Axis	AXIS_REF_SM3	Specify the axis, which is an instance of AXIS_REF_SM3

(2) Input variables

Input variable	Name	Type of data	Effective range	Initial value	Content
Enable	Effective	BOOL	TRUE, FALSE	FALSE	TRUE: Enable function block FALSE : Close the function block

(3) Output variables

Output variable	Name	Type of data	Effective range	Content

Valid	Axis status data can obtain flags	BOOL	TRUE, FALSE	TRUE: The output is valid
Busy	Function block is executing	BOOL	TRUE, FALSE	TRUE: The function block is running
Error	Error	BOOL	TRUE, FALSE	TRUE: The function block generates an exception and has stopped execution
ErrorID	Error code	SMC_ERROR	0	When an exception occurs, an error code is output
Torque	Actual axis torque read	LREAL		actual torque data of the axis read by the function block (in Nm)

(4) Conversion timing of output variables

Variable	When it becomes TRUE	When it becomes FALSE
Valid	✧ The output is valid	✧ Enable is TRUE, it is at the same time as FALSE of Enable ✧ Enable is FALSE, after 1 cycle
Busy	✧ Enable 's rising edge	✧ When Done is TRUE ✧ When Error is TRUE
Error	✧ When a fault occurs during the execution of a function block	✧ When the exception is resolved

Key points

- This function block can read the actual torque in the drive:

  When Enable is TRUE, the function block updates Torque data in real time;

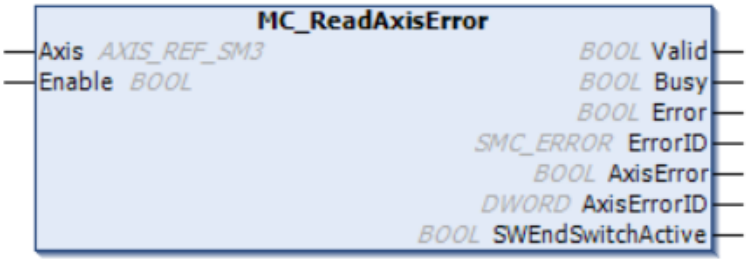
  When Enable is FALSE, the last updated data is retained.
- It is necessary to configure the communication data related to the actual torque to read the correct torque value.

➢ ※ Note: What this function block reads is the actual output torque value of the axis, not a percentage of the rated torque.

MC_ReadAxisError (FB)

Read the axis error status.

Name	MC_ReadAxisError (Read axis error status)		
Supported modes	CSP	CSV	CST
Graphical performance		ST performance _	




```
MC_ ReadAxisError (
Axis:=,
Enable:=,
Valid = >,
Busy = >,
Error = >,
ErrorID = >,
AxisError = >,
AxisErrorID = >,
SWEndSwitchActive => );
```

Variables

(1) Input and output variables

Input and output variable	Name	Type of data	Content
Axis	Axis	AXIS_REF_SM3	Specify the axis, which is an instance of AXIS_REF_SM3

(2) Input variables

Input variable	Name	Type of data	Effective range	Initial value	Content
Enable	Effective	BOOL	TRUE, FALSE	FALSE	TRUE: Enable function block FALSE : Close the function block

(3) Output variables

Output variable	Name	Type of data	Effective range	Content
Valid	Axis status data can obtain flags	BOOL	TRUE, FALSE	TRUE: The output is valid
Busy	Function block is executing	BOOL	TRUE, FALSE	TRUE: The function block is running
Error	Error	BOOL	TRUE, FALSE	TRUE: The function block generates an exception and has stopped execution
ErrorID	Error code	SMC_ERROR	0	When an exception occurs, an error code is output
AxisError	Axis error flag	BOOL	TRUE, FALSE	TRUE when an axis error occurs
AxisErrorID	Axis error code	DWORD		Read the error code of the axis
SWEndSwitchActive	Software limit switch is valid	BOOL	TRUE, FALSE	TRUE: soft limit trigger

(4) Conversion timing of output variables

Variable	When it becomes TRUE	When it becomes FALSE
Valid	✧ The output is valid	✧ Enable is TRUE, it is at the same time as FALSE of Enable ✧ Enable is FALSE, after 1 cycle
Busy	✧ Enable 's rising edge	✧ When Done is TRUE ✧ When Error is TRUE
Error	✧ When a fault occurs during the execution of a function block	✧ When the exception is resolved

Key points

- This function block can read the error code in the drive:

  When Enable is TRUE, the function block updates the data related to axis error in real time;

  When Enable is FALSE, the last updated data is retained.

MC_ReadParameter (FB)

Read the parameters of the axis.

Name	MC_ReadParameter (Read the parameters of the axis)		
Supported modes	CSP	CSV	CST
Graphical performance		ST performance _	
		MC_ ReadParameter (Axis:=, Enable:=, ParameterNumber :=, Valid = >, Busy = >, Error = >, ErrorID = >, Value=>);	

Variables

(1) Input and output variables

Input and output variable	Name	Type of data	Content
Axis	Axis	AXIS_REF_SM3	Specify the axis, which is an instance of AXIS_REF_SM3

(2) Input variables

Input variable	Name	Type of data	Effective range	Initial value	Content
Enable	Effective	BOOL	TRUE, FALSE	FALSE	TRUE: Enable function block FALSE : Close the function block
ParameterNumber	Axis parameter number	DINT	-	0	The serial number corresponding to the axis parameter, ParameterNumber (axis parameter serial number) can view AXIS_REF_SM3, and can also read the parameters defined in the corresponding bus, such as the status word in the COE

(3) Output variables

Output variable	Name	Type of data	Effective range	Content
Valid	Axis parameter data can obtain flags	BOOL	TRUE, FALSE	TRUE : can obtain axis parameter data
Busy	Function block is executing	BOOL	TRUE, FALSE	TRUE: The function block is running
Error	Error	BOOL	TRUE, FALSE	TRUE: The function block generates an exception and has stopped execution
ErrorID	Error code	SMC_ERROR	0	When an exception occurs, an error code is output
Value	Axis parameters read	LREAL	-	Axis parameters read by the function block

(4) Conversion timing of output variables

Variable	When it becomes TRUE	When it becomes FALSE
Valid	✧ The output is valid	✧ Enable is TRUE, it is at the same time as FALSE of Enable ✧ Enable is FALSE, after 1 cycle
Busy	✧ Enable 's rising edge	✧ When Done is TRUE ✧ When Error is TRUE
Error	✧ When a fault occurs during the execution of a function block	✧ When the exception is resolved

Key points

This function block can read the parameters corresponding to ParameterNumber (axis parameter number) when Enable is TRUE, and can also read the parameters defined in the fieldbus protocol, such as the 16# 6041 status word in the HCFA X3EB drive.

➢ ※ Note: When reading bus protocol parameters, you need to do the following operations in the program

diParameterNumeber is declared as a DINT type variable and added to the input pin ParameterNumeber of the function block.

usiDataLength (write the length of the data in the object dictionary you want to read) is declared as a variable of type USINT.

uiIndex (write the index in the object dictionary you want to read) is declared as a UINT type variable.

usisubIndex (write the sub-index in the object dictionary you want to read) is declared as a DINT type variable.

For example, read the return mode in the HCFA drive object dictionary:

```
diParameterNumeber : =-DWORD_TO_DINT(SHL(USINT_TO_DWORD( 1 ), 24) +SHL(UINT_TO_DWORD( 16 # 6098 ), 8 )+ 16 # 00 );
```

MC_ReadBoolParameter (FB)

Read the BOOL type parameters of the axis.

Name	MC_ReadBoolParameter (Read the BOOL parameter of the axis)		
Supported modes	CSP	CSV	CST
Graphical performance		ST performance _	


```
MC_ ReadBoolParameter (
Axis:=,
Enable:=,
ParameterNumber :=,
Valid = >,
Busy = >,
Error = >,
ErrorID = >,
Value=> );
```

Variables

(1) Input and output variables

Input and output variable	Name	Type of data	Content
Axis	Axis	AXIS_REF_SM3	Specify the axis, which is an instance of AXIS_REF_SM3

(2) Input variables

Input variable	Name	Type of data	Effective range	Initial value	Content
Enable	Effective	BOOL	TRUE, FALSE	FALSE	TRUE: Enable function block FALSE : Close the function block
ParameterNumber	Axis parameter number	DINT		0	The serial number corresponding to the axis parameter, ParameterNumber (axis parameter serial number) can view AXIS_REF_SM3, and can also read the parameters defined in the corresponding bus, such as the status word in the COE

(3) Output variables

Output variable	Name	Type of data	Effective range	Content
Valid	Axis parameter data can obtain flags	BOOL	TRUE, FALSE	TRUE : can obtain axis parameter data
Busy	Function block is executing	BOOL	TRUE, FALSE	TRUE: The function block is running
Error	Error	BOOL	TRUE, FALSE	TRUE: The function block generates an exception and has stopped execution
ErrorID	Error code	SMC_ERROR	0	When an exception occurs, an error code is output
Value	Axis BOOL parameter read	BOOL	TRUE, FALSE	Axis parameters read by the function block

(4) Conversion timing of output variables

Variable	When it becomes TRUE	When it becomes FALSE
Valid	✧ The output is valid	✧ Enable is TRUE, it is at the same time as FALSE of Enable ✧ Enable is FALSE, after 1 cycle
Busy	✧ Enable 's rising edge	✧ When Done is TRUE ✧ When Error is TRUE
Error	✧ When a fault occurs during the execution of a function block	✧ When the exception is resolved

Key points

When Enable is TRUE, this function block can read parameters corresponding to ParameterNumber or parameters defined in fieldbus protocols, such as the BOOL values of DI and DO in group 16#2104 of HCFA X3EB drive.

➢ ※ Note: When reading bus protocol parameters, you need to do the following operations in the program

diParameterNumeber is declared as a DINT type variable and added to the input pin ParameterNumeber of the function block.

usiDataLength (write the length of the data in the object dictionary you want to read) is declared as a variable of type USINT.

uiIndex (write the index in the object dictionary you want to read) is declared as a UINT type variable.

usisubIndex (write the sub-index in the object dictionary you want to read) is declared as a DINT type variable.

diParameterNumeber : =-DWORD_TO_DINT(SHL(USINT_TO_DWORD(usidatalength),24)+SHL(UINT_TO_DWORD(uiindex),8)+usisubindex);

For details, please refer to MC_ReadParameter to read the bus protocol parameters.

MC_WriteParameter (FB)

Modify the parameters of the axis.

Name	MC_WriteParameter (Modify the parameters of the axis)		
Supported modes	CSP	CSV	CST
Graphical performance		ST performance _	
		MC_ WriteParameter (Axis:=, Execute:=, ParameterNumber :=, Value:=, Done= >, Busy = >, Error = >, ErrorID =>);	

Variables

(1) Input and output variables

Input and output variable	Name	Type of data	Content
Axis	axis	AXIS_REF_SM3	Specify the axis, which is an instance of AXIS_REF_SM3

(2) Input variables

Input variable	Name	Type of data	Effective range	Initial value	Content
Execute	Effective	BOOL	TRUE, FALSE	FALSE	TRUE: Enable function block FALSE : Close the function block
ParameterNumber	Axis parameter number	DINT	-	0	Enter the index and sub-index of the axis parameter you want to modify
Value	Parameter setting	LREAL	-	-	Set parameter value

(3) Output variables

Output variable	Name	Type of data	Effective range	Content
D one	Parameter setting succeeded	BOOL	TRUE, FALSE	TRUE : Parameter setting is successful

Busy	Function block is executing	BOOL	TRUE, FALSE	TRUE: The function block is running
Error	Error	BOOL	TRUE, FALSE	TRUE: The function block generates an exception and has stopped execution
ErrorID	Error code	SMC_ERROR	0	When an exception occurs, an error code is output

(4) Conversion timing of output variables

Variable	When it becomes TRUE	When it becomes FALSE
D one	✧ Set successfully	✧ Execute is TRUE, it is at the same time as FALSE of Execute ✧ Execute is FALSE, after 1 cycle
Busy	✧ Rising edge of Execute	✧ When Done is TRUE ✧ When Error is TRUE
Error	✧ When a fault occurs during the execution of a function block	✧ When the exception is resolved

Key points

- The rising edge of Execute is to start the function block. The rising edge can be triggered again during the execution of the function block. Each rising edge will reload the input parameters of the function block and execute the function block again.
- This function block can modify the parameters corresponding to ParameterNumber (axis parameter number), and can also modify the parameters defined in the fieldbus protocol, such as the 16# 6 041 status word in the HCFA X3EB drive.

➢ ※**Note:** When modifying the bus protocol parameters, you need to do the following operations in the program

diParameterNumeber is declared as a DINT type variable and added to the input pin ParameterNumeber of the function block.

usiDataLength (write the length of the data in the object dictionary you want to read) is declared as a variable of type USINT.

uiIndex (write the index in the object dictionary you want to read) is declared as a UINT type variable.

usisubIndex (write the sub-index in the object dictionary you want to read) is declared as a DINT type variable. diParameterNumeber : =-DWORD_TO_DINT(SHL(USINT_TO_DWORD(usidatalength),24)+SHL(UINT_TO_DWORD(uiindex),8)+usisubindex);

For example, modify the return mode in the HCFA drive object dictionary:

diParameterNumeber : =-DWORD_TO_DINT(SHL(USINT_TO_DWORD(1), 24) +SHL(UINT_TO_DWORD(16 # 6098), 8)+ 16 # 00);

MC_WriteBOOLParameter (FB)

OOL type parameter of the axis.

Name	MC_WriteBoolParameter (Modify the BOOL parameter of the axis)		
Supported modes	CSP	CSV	CST
Graphical performance		ST performance _	
		MC_ WriteBoolParameter (Axis:=, Execute:=, ParameterNumber :=, Value:=, Done= >, Busy = >, Error = >, ErrorID =>);	

Variables

(1) Input and output variables

Input and output variable	Name	Type of data	Content
Axis	Axis	AXIS_REF_SM3	Specify the axis, which is an instance of AXIS_REF_SM3

(2) Input variables

Input variable	Name	Type of data	Effective range	Initial value	Content
Execute	Effective	BOOL	TRUE, FALSE	FALSE	TRUE: Enable function block FALSE : Close the function block
ParameterNumber	Axis parameter number	DINT		0	Enter the index and sub-index of the axis parameter you want to modify
Value	Parameter setting	BOOL	TRUE, FALSE		Set parameter value

(3) Output variables

Output variable	Name	Type of data	Effective range	Content
Done	Parameter setting succeeded	BOOL	TRUE, FALSE	TRUE : Parameter setting is successful
Busy	Function block is executing	BOOL	TRUE, FALSE	TRUE: The function block is running
Error	Error	BOOL	TRUE, FALSE	TRUE: The function block generates an exception and has stopped execution
ErrorID	Error code	SMC_ERROR	0	When an exception occurs, an error code is output

(4) Conversion timing of output variables

Variable	When it becomes TRUE	When it becomes FALSE
Done	✧ Set successfully	✧ Execute is TRUE, it is at the same time as FALSE of Execute ✧ Execute is FALSE, after 1 cycle
Busy	✧ Rising edge of Execute	✧ When Done is TRUE ✧ When Error is TRUE
Error	✧ When a fault occurs during the execution of a function block	✧ When the exception is resolved

Key points

- The rising edge of Execute is to start the function block. The rising edge can be triggered again during the execution of the function block. Each rising edge will reload the input parameters of the function block and execute the function block again.
- This function block can modify the parameters corresponding to ParameterNumber (axis parameter number), and can also modify the parameters defined in the fieldbus protocol, such as the 16#6041 status word in the HCFA X3EB drive.

➢ ※ Note: When modifying the bus protocol parameters, you need to do the following operations in the program

diParameterNumeber is declared as a DINT type variable and added to the input pin ParameterNumeber of the function block.

usiDataLength (write the length of the data in the object dictionary you want to read) is declared as a variable of type USINT.

uiIndex (write the index in the object dictionary you want to read) is declared as a UINT type variable.

usisubIndex (write the sub-index in the object dictionary you want to read) is declared as a DINT type variable. diParameterNumeber : =-

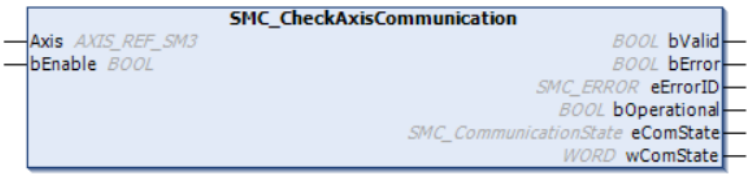
DWORD_TO_DINT(SHL(USINT_TO_DWORD(usidatalength),24)+SHL(UINT_TO_DWORD(uiindex),8)+usisubindex);

specific operations, please refer to MC_ReadParameter to read the bus protocol parameters.

SMC_CheckAxisCommunication (FB)

Get the current communication status of the axis.

Name	SMC_CheckAxisCommunication (Get the current communication status of the axis)		
Supported modes	CSP	CSV	CST
Graphical performance		ST performance _	




```
SMC_ CheckAxisCommunication (
Axis:=,
bEnable :=,
bValid = >,
bError = >,
eErrorID = >,
bOperational = >,
eComState = >,
wComState => );
```

Variables

(1) Input and output variables

Input and output variable	Name	Type of data	Content
Axis	Axis	AXIS_REF_SM3	Specify the axis, which is an instance of AXIS_REF_SM3

(2) Input variables

Input variable	Name	Type of data	Effective range	Initial value	Content
bEnable	Effective	BOOL	TRUE, FALSE	FALSE	TRUE: Enable function block FALSE : Close the function block

(3) Output variables

Output variable	Name	Type of data	Effective range	Content
bValid	Axis parameter data can obtain flags	BOOL	TRUE, FALSE	TRUE: can obtain axis parameter data
bError	Error	BOOL	TRUE, FALSE	TRUE: The function block generates an exception and has stopped execution
bErrorID	Error code	SMC_ERROR	0	When an exception occurs, an error code is output
bOperational	Communication normal sign	BOOL	TRUE, FALSE	TRUE: The communication is normal and the axis can be operated
eComState	Communication status	SMC _CommunicationState		Communication state, refer to SMC _CommunicationState
wComState	Communication code	WORD		Used to distinguish the current communication status, Same as axis structure variable wCommunicationState

(4) Conversion timing of output variables

Variable	When it becomes TRUE	When it becomes FALSE
bValid	✧ The output is valid	✧ bEnable is TRUE, it is at the same time as the FALSE of bEnable ✧ bEnable is FALSE, after 1 cycle
bError	✧ When a fault occurs during the execution of a function block	✧ When the exception is resolved

Key points

This function block can obtain the communication status of the axis:

- When bEnable is TRUE, the function block updates the communication status data in real time.

  When bEnable is FALSE, the last updated data is retained.

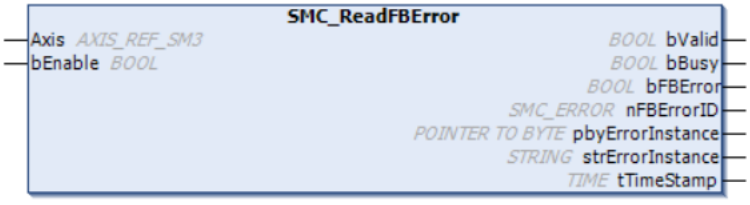
- wComState = 100, b Operational is TRUE, axis communication is normal, and other function block operations can be performed.

  When wComState <100, the axis communication is being initialized, and the communication has not been completed yet, and the axis cannot run.

  When wComState >100, the axis communication fails and cannot be restored by MC _Reset. After the bus connection is re-established, the S MC3_ReInitDrive function block must be called to reinitialize the axis.

SMC_ReadFBError (FB)

Error reading MC and SMC function blocks.

Name	SMC_ReadFBError (read function block error)		
Supported modes	CSP	CSV	CST
Graphical performance		ST performance _	
		SMC_ ReadFBError (Axis:=, bEnable :=, bValid = >, bBusy = >, bFBError = >, nFBErrorID = >, pbyErrorInstance = >, strErrorInstance = >, tTimeStamp =>);	

Variables

(1) Input and output variables

Input and output variable	Name	Type of data	Content
Axis	Axis	AXIS_REF_SM3	Specify the axis, which is an instance of AXIS_REF_SM3

(2) Input variables

Input variable	Name	Type of data	Effective range	Initial value	Content
bEnable	Effective	BOOL	TRUE, FALSE	FALSE	TRUE: Enable function block FALSE : Close the function block

(3) Output variables

Output variable	Name	Type of data	Effective range	Content
bValid	Axis parameter data can obtain flags	BOOL	TRUE, FALSE	When the axis parameter data can be obtained, it becomes TRUE
bBusy	Function block is executing	BOOL	TRUE, FALSE	TRUE: The function block is running
bFBError	Error	BOOL	TRUE, FALSE	TRUE: An abnormality occurred in a function block
nFBErrorID	Error code	SMC_ERROR	0	When an exception occurs, an error code is output
pbyErrorInstance	Error instance	POINTER TO BYTE		Point to the function block reporting the error
strErrorInstance	Error instance	STRING		Path and Name of abnormal function block
tTimeStamp	Time of error	TIME		The time stamp when the function block exception occurred

(4) Conversion timing of output variables

Variable	When it becomes TRUE	When it becomes FALSE
bValid	✧ The output is valid	✧ bEnable is TRUE, it is at the same time as the FALSE of bEnable ✧ bEnable is FALSE, after 1 cycle
bBusy	✧ bEnable 's rising edge	✧ bDone is TRUE ✧ bWhen Error is TRUE
bFBError	✧ When a fault occurs during the execution of a function block	✧ When the exception is resolved

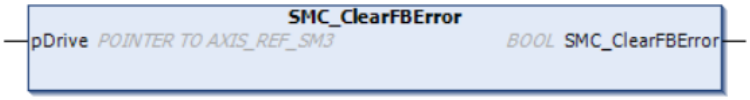
Key points

- Enable is set to TRUE, SMC_ReadFBError enters the execution state. At this time, the function block detects whether the function block using this axis reports an error.
- setting Enable to FALSE, SMC_Read FBError will no longer read the function block error status, and will retain the previously updated function block status information. If the function block reports an error when Enable is FALSE, when Enable is set to TRUE, the axis error message will be displayed, but when Enable is reset to FALSE, the output error message will be reset.
- clear the error information of the function block read by SMC_ReadFBError, you need to use SMC_ClearFBError. Refer to SMC_ClearFBError for For details HYPERLINK \l "SMC_ClearFBError（FUN）".

➢ ※ Note: If multiple function blocks of the same Axis report an error, SMC_ReadFBError will only save the information of the first function block that reports an error.

SMC_ClearFBError (FUN)

Clear MC and SMC function block errors.

Name	SMC_ClearFBError (Clear function block error)		
Supported modes	CSP	CSV	CST
Graphical performance		ST performance _	
		SMC_ ClearFBError (pDrive :=ADR());	

Variables

Input variables

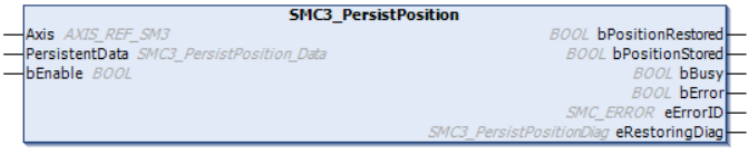
Input variable	Name	Type of data	Content
pDrive	Pointing axis	POINTER TO AXIS_REF	Pointer points to AXIS_REF_SM3

Key points

- This function block is a function block exception reset function, which is called when the function block exception needs to be reset.
- pDrive pin needs to take the address of AXIS_REF_SM3 type variable.

SMC3_PersistPosition (FB)

Save and record the position of the absolute encoder on the real axis (multi-turn). Virtual axis and logical axis are not supported.

Name	SMC_ PersistPosition (save position)		
Supported modes	CSP	CSV	CST
Graphical performance		ST performance _	
		SMC3_ PersistPosition(Axis:=, PersistentData :=, bEnable :=, bPositionRestored = >, bPositionStored = >, bBusy = >,	

	bError = >, eErrorID = >, eRestoringDiag =>);

Variables

(1) Input and output variables

Input and output variable	Name	Type of data	Content
Axis	Axis	AXIS_REF_SM3	Specify the axis, which is an instance of AXIS_REF_SM3
PersistentData	Keep data	SMC3_PersistPosition_Data	For location saving

(2) Input variables

Input variable	Name	Type of data	Effective range	Initial value	Content
bEnable	Effective	BOOL	TRUE, FALSE	FALSE	TRUE: Enable function block FALSE : Close the function block

(3) Output variable

Output variable	Name	Type of data	Effective range	Content
bPositionRestored	Location recovery flag	BOOL	TRUE, FALSE	TRUE: The position saved before the axis is powered off is restored successfully
bPositionStored	Location save flag	BOOL	TRUE, FALSE	TRUE: The current position of the axis has been saved
bBusy	Function block is executing	BOOL	TRUE, FALSE	TRUE: The function block is running
bError	Error	BOOL	TRUE, FALSE	TRUE: The function block generates an exception and has stopped execution
nErrorID	error code	SMC_ERROR	0	When an exception occurs, an error code is output
eRestoringDiag	Restore diagnostic information	SMC3_PersistPositionDiag	SMC3_PPD_RESTORING_OK	The diagnostic information of the function block, refer to SMC3_PersistPositionDiag for the specific meaning

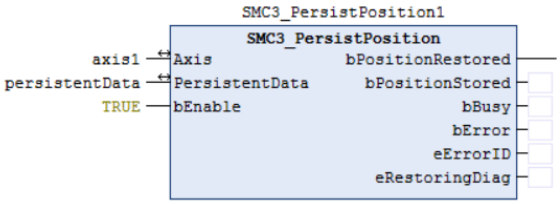
Key points

- This function block is only used for real axes, and it must be executed first after the controller is powered on. It is recommended that the Enable pin of the function block is TRUE by default.
- If the output variable bPositionRestored is TRUE, it means that the function block loads data correctly after power-on.
- This function block can keep the data of Axis.iTurn and Axis.PosOffsetForResiduals after power-off. It is necessary to configure SMC 3_PersistPosition_Data as a power-off retentive variable.
- When the function block restores the position data after power-on, the axis scaling and other parameters must be the same as before the power-off, so when this function block is used together with S MC_ChangeGearingRatio , there may be problems, and you need to pay attention to the execution sequence.
- SMC 3_PersistPosition_Data is configured as a retentive variable tutorial:

First declare an instance of SMC 3_P e rsistPosition_Data type p ersistentData, and the variable type is R ETAIN PERSISTENT.

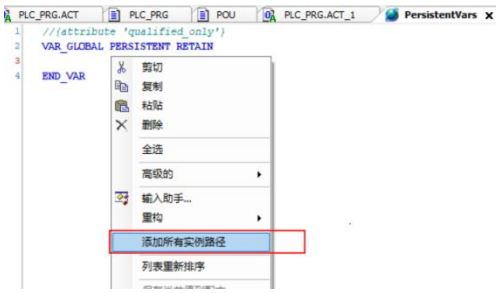
```
VAR PERSISTENT RETAIN
    persistentData: SMC3_PersistPosition_Data;
END_VAR
```

Add it to the PersistentData pin of SMC 3_P e rsistPosition.



Create a P e rsistentVars variable (persistent variable)

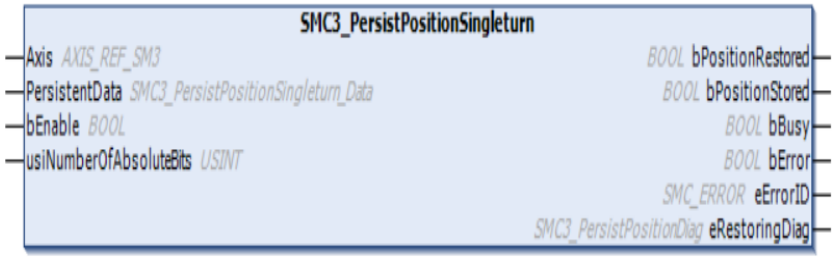
Recompile the program, right-click on the blank space of " P e rsistentVars " after compiling without error, and choose to add all instance paths.



SMC3_PersistPositionSingleturn (FB)

This function block is used to save and record the position of the real axis absolute encoder (single turn).

Name	SMC_PersistPositionS ing leturn (save position)		
Supported modes	CSP	CSV	CST
Graphical performance		ST performance _	




```
SMC3_ PersistPositionSingleturn(
Axis:=,
PersistentData :=,
bEnable :=,
usiNumberOfAbsoluteBits :=,
bPositionRestored = >,
bPositionStored = >,
bBusy = >,
bError = >,
eErrorID = >,
eRestoringDiag => );
```

Variables

(1) Input and output variables

Input and output variable	Name	Type of data	Content
Axis	Axis	AXIS_REF_SM3	Specify the axis, which is an instance of AXIS_REF_SM3
PersistentData	Keep data	SMC3_PersistPositionSingleturn_Data	For location saving

(2) Input variables

Input variable	Name	Type of data	Effective range	Initial value	Content
bEnable	Effective	BOOL	TRUE, FALSE	FALSE	TRUE: Enable function block FALSE : Close the function block
usiNumberOfAbsoluteBits	Absolute encoder data bit	USINT		16	Absolute encoder data bits (8- 32)

(3) Output variables

Output variable	Name	Type of data	Effective range	Content
bPositionRestored	Location recovery flag	BOOL	TRUE, FALSE	TRUE: The position saved before the axis is powered off is restored successfully
bPositionStored	Location save flag	BOOL	TRUE, FALSE	TRUE: The current position of the axis has been saved
bBusy	Function block is executing	BOOL	TRUE, FALSE	TRUE: The function block is running
bError	Error	BOOL	TRUE, FALSE	TRUE: The function block generates an exception and has stopped execution
nErrorID	error code	SMC_ERROR	0	When an exception occurs, an error code is output
eRestoringDiag	Restore diagnostic	SMC3_PersistPositionDiag	SMC3_PPD_RESTORING_OK	dynamic diagnostic information of the loading position, refer to SMC3_PersistPositionDiag

	information			

Key points

- This function block is only used for real axis, and it must be executed first after the controller is powered on. It is recommended that the Enable pin of the function block is TRUE by default
- Output variable bPositionRestored is TRUE, it means that the function block loads the data correctly after power-on.
- When the function block restores the position data after power-on, the axis scaling and other parameters must be the same as before the power-off, so when this function block is used together with S MC_ChangeGearingRatio , there may be problems, and you need to pay attention to the execution sequence.

➢ ※ Note: Refer to S MC_ChangeGearingRatio .

SMC_ChangeGearingRatio (FB)

Modify the electronic gear ratio and axis type.

Name	SMC_ChangeGearingRatio (Modify the electronic gear ratio)		
Supported modes	CSP	CSV	CST
Graphical performance		ST performance _	


```
SMC_ ChangeGearingRatio (
Axis:=,
bExecute :=,
dwRatioTechUnitsDenom :=,
iRatioTechUnitsNum :=,
fPositionPeriod :=,
iMovementType :=,
bDone = >,
bBusy = >,
bError = >,
nErrorID => );
```

Variables

(1) Input and output variables

Input and output variable	Name	Type of data	Content
Axis	Axis	AXIS_REF_SM3	Specify the axis, which is an instance of AXIS_REF_SM3

(2) Input variables

Input variable	Name	Type of data	Effective range	Initial value	Content

bExecute	Start up	BOOL	TRUE, FALSE	FALSE	TRUE: Enable function block
dwRatioTechUnitsDenom		DWORD		0	The denominator of the conversion factor that converts the drive increment to the command unit
iRatioTechUnitsNum		DINT		0	The numerator of the conversion factor that converts drive increments to instruction units
fPositionPeriod	Position cycle	LREAL		0	Only useful for modal axis, position cycle
iMovementType	Axis type selection	S MC_MOVEMENTTYPE	0, 1	rotary	0: rotary modal axis 1 : linear axis

(3) Output variables

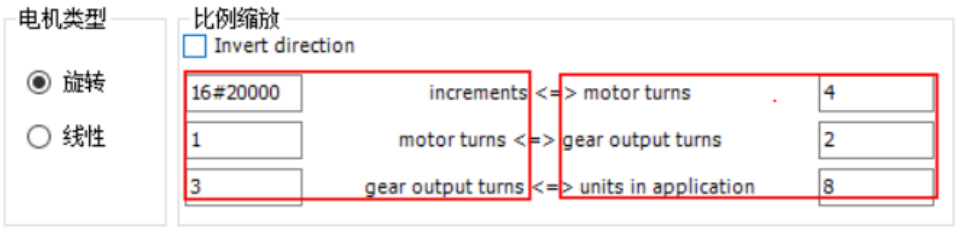
Output variable	Name	Type of data	Effective range	Content
bDone	Finish	BOOL	TRUE, FALSE	TRUE: The execution of the function block is complete
bBusy	Function block is executing	BOOL	TRUE, FALSE	TRUE: The function block is running
bError	Error	BOOL	TRUE, FALSE	TRUE: The function block generates an exception and has stopped execution
nErrorID	error code	SMC_ERROR	0	When an exception occurs, an error code is output

(4) Conversion timing of output variables

variable	When it becomes TRUE	When it becomes FALSE
bDone	✧ When reset is complete	✧ bExecute is TRUE at the same time as the FALSE of bExecute ✧ bExecute is FALSE, after 1 cycle
bBusy	✧ bExecute rising edge	✧ bDone is TRUE ✧ bError is TRUE
bError	✧ When a fault occurs during the execution of a function block	✧ When the exception is resolved

Key points

- The rising edge of bExecute is to start the function block. The rising edge can be triggered again during the execution of the function block. The input parameters of the function block will be reloaded each time the rising edge is executed, and the function block will be executed again.
- The gear ratio and axis type of the axis can be modified through this function block. After b Done is TRUE, it may be necessary to restart the axis through S MC3_ReInitDrive to ensure that all variables are successfully initialized.
- The following is a brief description of dwRatioTechUnitsDenom and iRatioTechUnitsNum :



dwRatioTechUnitsDenom is multiplied by the three parameters on the left, and iRatioTechUnitsNum is multiplied by the

three parameters on the right. Please note that if the calculated dwRatioTechUnitsDenom and iRatioTechUnitsNum can be divided, the software will automatically perform the division.

dwRatioTechUnitsDenom := increments (1 6 # 20000) * motor turns (1) * gear output turns (3);

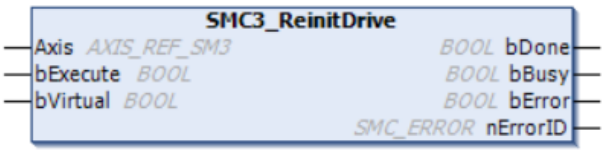
iRatioTechUnitsNum ： = motor turns (4) * gear output turns (2) * units in application (8);

dwRatioTechUnitsDenom	DWORD	6144			Parameter number 1051
iRatioTechUnitsNum	DINT	1			Parameter number 1052

The obtained parameters can be viewed in the axis structure (A XIS_REF_SMC3).

SMC3_ReInitDrive (FB)

Restart the axis.

Name	SMC3_ReInitDrive (restart axis)		
Supported modes	CSP	CSV	CST
Graphical performance		ST performance _	
		SMC3_ ReinitDrive(Axis:=, bExecute :=, bVirtual :=, bDone = >, bBusy = >, bError = >, nErrorID =>);	

Variables

(1) Input and output variables

Input and output variable	Name	Type of data	Content
Axis	Axis	AXIS_REF_SM3	Specify the axis, which is an instance of AXIS_REF_SM3

(2) Input variables

Input variable	Name	Type of data	Effective range	Initial value	Content
bExecute	Start up	BOOL	TRUE, FALSE	FALSE	TRUE: Enable function block
bVirtual	virtual	BOOL	TRUE, FALSE	FALSE	TRUE: virtual axis

(3) Output variables

Output variable	Name	Type of data	Effective range	Content

bDone	Finish	BOOL	TRUE, FALSE	TRUE: The execution of the function block is complete
bBusy	Function block is executing	BOOL	TRUE, FALSE	TRUE: The function block is running
bError	Error	BOOL	TRUE, FALSE	TRUE: The function block generates an exception and has stopped execution
nErrorID	error code	SMC_ERROR	0	When an exception occurs, an error code is output

(4) Conversion timing of output variables

Variable	When it becomes TRUE	When it becomes FALSE
bDone	When reset is complete	bExecute is TRUE at the same time as the FALSE of bExecute b Execute is FALSE, after 1 cycle
bBusy	bExecute rising edge	bDone is TRUE bError is TRUE
bError	When a fault occurs during the execution of a function block	When the exception is resolved

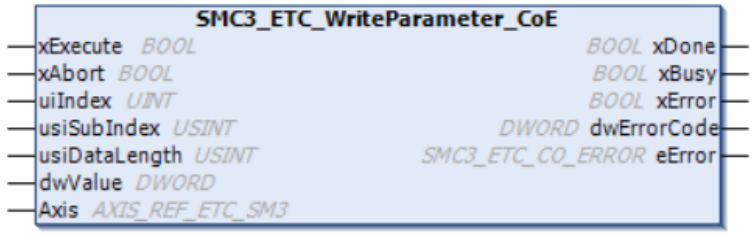
Key points

- When bExecute is TRUE, the axis will be reinitialized, and the axis cannot perform any operations until the initialization is completed.
- If bVirtual is TRUE, the axis will be judged as a virtual axis and will exist as a virtual axis. It has no effect on the fieldbus device and can operate normally, but it will not communicate with the real device.
- The axis status of the function block is power-off when it’ s running.

Axis status before function block operation	Axis status after the function block is completed	Remark
Power_Off	Power_Off	
Errorstop	Power_Off	
Standstill	Standstill	
Homing	Standstill	function block (SMC3_ReInitDrive) Execute is TRUE, the output pin Done of MC_Home is directly TRUE
Continuous Motion	Standstill	The original motion control function block will report an error error Content is: " SMC_AXIS_NOT_READY_FOR_MOTION "
Synchronized Motion	Standstill	
Discrete Motion	Standstill	
Stopping	Standstill	MC_Stop will report an error "SMC_REGULATOR_OR_START_NOT_SET"

SMC3_ETC_WriteParameter_CoE (FB)

Modify EtherCAT parameters.

Name	SMC3_ETC_WriteParameter_CoE (Modify Ether CAT parameters)		
Supported modes	CSP	CSV	CST
Graphical performance		ST performance _	
		SMC3_ETC_WriteParameter_ CoE(xExecute :=, xAbort :=, uiIndex :=, usiSubIndex :=, usiDataLength :=, dwValue :=, Axis:=, xDone = >, xBusy = >, xError = >, dwErrorCode = >, eError =>);	

Variables

(1) Input and output variables

Input and output variable	Name	Type of data	Content
Axis	Axis	AXIS_REF_SM3	Specify the axis, which is an instance of AXIS_REF_SM3

(2) Input variables

Input variable	Name	Type of data	Effective range	Initial value	Content
xExecute	Start up	BOOL	TRUE, FALSE	FALSE	TRUE: Enable function block
xAbort	Interrupt	BOOL	TRUE, FALSE	FALSE	TRUE: interrupt function block
uiIndex	index	UINT		0	Index of the object dictionary
usiSubIndex	Subindex	USINT		0	Sub-index of the object dictionary
usiDataLength	Data length	USINT		0	Data length of the object dictionary
dwValue	Parameter setting	DWORD		0	Modify value

(3) Output variables

Output variable	Name	Type of data	Effective range	Content
xDone	Parameter setting succeeded	BOOL	TRUE, FALSE	TRUE: The execution of the function block is complete
xBusy	Function block is executing	BOOL	TRUE, FALSE	TRUE: The function block is running
xError	Error	BOOL	TRUE, FALSE	TRUE: The function block generates an exception and has stopped execution
dwErrorCode	Error code	DWORD	0	An error occurred when converting S DO through C AN open
eError	Error	S MC3_ETC_CO_ERROR		Function block alarm

(4) Conversion timing of output variables

Variable	When it becomes TRUE	When it becomes FALSE
xDone	✧ When reset is complete	✧ xExecute is TRUE, it is at the same time as the FALSE of xExecute ✧ xExecute is FALSE, after 1 cycle
xBusy	✧ xExecute rising edge	✧ xDone is TRUE ✧ bError is TRUE
xError	✧ When a fault occurs during the execution of a function block	✧ When the exception is resolved

Key points

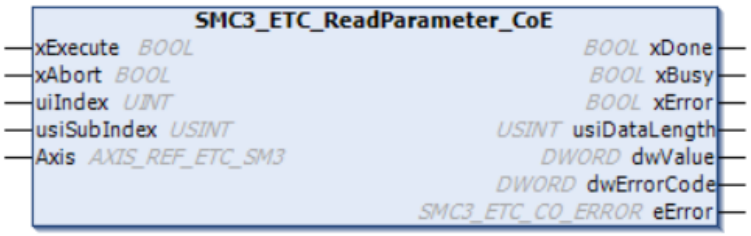
- xExecute is to start the function block. The rising edge can be triggered again during the execution of the function block. The input parameters of the function block will be reloaded every time the rising edge is executed, and the function block will be executed again.
- This function block can modify the parameters defined in the fieldbus protocol. Take the modified object dictionary 1 6 # 6098 (return to original mode) as an example, uiIndex (index) is assigned to 1 6 # 6098, usiSubIndex (sub-index) is assigned to 1 6 # 00, the value of usiDataLength (data length) is 1, and the value of dwValue (setting parameter) is 30. At

this time, xExecute is set to TRUE, and when the function block xDone becomes TRUE, the original mode is successfully modified.

SMC3_ETC_ReadParameter_CoE (FB)

Get EtherCAT parameters.

Name	SMC3_ETC_ReadParameter_CoE (Get Ether CAT parameters)		
Supported modes	CSP	CSV	CST
Graphical performance		ST performance _	




```
SMC3_ETC_ReadParameter_ CoE(
xExecute :=,
xAbort :=,
uiIndex :=,
usiSubIndex :=,
Axis:=,
xDone = >,
xBusy = >,
xError = >,
usiDataLength = >,
dwValue = >,
dwErrorCode = >,
eError => );
```

Variables

(1) Input and output variables

Input and output variable	Name	Type of data	Content
Axis	Axis	AXIS_REF_SM3	Specify the axis, which is an instance of AXIS_REF_SM3

(2) Input variables

Input variable	Name	Type of data	Effective range	Initial value	Content
xExecute	Start up	BOOL	TRUE, FALSE	FALSE	TRUE: Enable function block
xAbort	Interrupt	BOOL	TRUE, FALSE	FALSE	TRUE: interrupt function block
uiIndex	index	UINT		0	Index of the object dictionary
usiSubIndex	Subindex	USINT		0	Sub-index of the object dictionary

(3) Output variables

Output variable	Name	Type of data	Effective range	Content
xDone	Parameter reading success flag	BOOL	TRUE, FALSE	TRUE: The execution of the function block is complete
xBusy	Function block is executing	BOOL	TRUE, FALSE	TRUE: The function block is running
xError	Error	BOOL	TRUE, FALSE	TRUE: The function block generates an exception and has stopped execution
dwErrorCode	Error code	DWORD	0	An error occurred when converting S DO through C AN open

eError	Error	S MC3_ETC_CO_ERROR		Function block alarm
usiDataLength	Data length	USINT		Data length of the object dictionary
dwValue	Parameter setting	DWORD		Read value

(4) Conversion timing of output variables

Variable	When it becomes TRUE	When it becomes FALSE
xDone	✧ When reset is complete	✧ xExecute is TRUE, it is at the same time as the FALSE of xExecute ✧ xExecute is FALSE, after 1 cycle
xBusy	✧ xExecute rising edge	✧ xDone is TRUE ✧ xError is TRUE
xError	✧ When a fault occurs during the execution of a function block	✧ When the exception is resolved

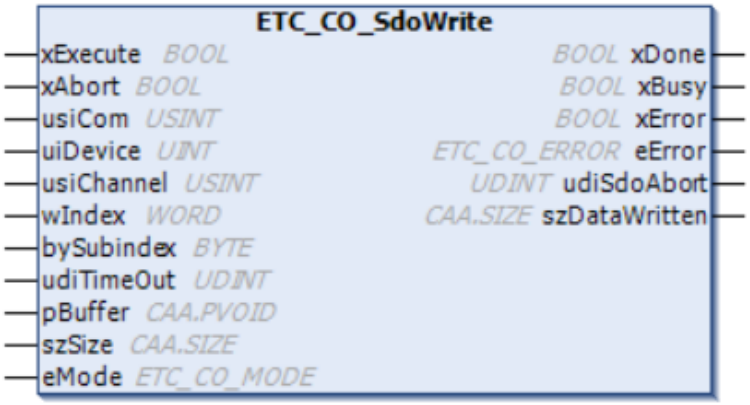
Key points

- xExecute is to start the function block. The rising edge can be triggered again during the execution of the function block. The input parameters of the function block will be reloaded every time the rising edge is executed, and the function block will be executed again.
- This function block can read the parameters defined in the fieldbus protocol. Take reading object dictionary 1 6 # 6080 (maximum motor speed) as an example, uiIndex (index) is assigned as 1 6 # 6080, usiSubIndex (sub-index) is assigned as 1. 6 # 00, at this time bExecute is set to TRUE, when the function block bDone becomes TRUE, the maximum motor speed is successfully read, and usiDataLength (data length) and dwValue (numerical value) display the current parameter value.

ETC_CO_SdoWrite (FB)

Modify the slave parameters.

Name	ETC_CO_SdoWrite (modify slave parameters)		
Supported modes	CSP	CSV	CST
Graphical performance		ST performance _	




```
ETC_CO_ SdoWrite (
xExecute :=,
xAbort :=,
usiCom :=,
uiDevice :=,
Channel : =,
wIndex : =,
bySubindex : =,
udiTimeOut : =,
pBuffer : =,
szSize : =,
eMode : =,
xDone = >,
xBusy = >,
xError = >,
eError = >,
udiSdoAbort = >,
szDataWritten =>);
```

Variables

(1) Input variables

Input variable	Name	Type of data	Effective range	Initial value	Content
xExecute	Start up	BOOL	TRUE, FALSE	FALSE	TRUE: Enable function block
xAbort	Interrupt	BOOL	TRUE, FALSE	FALSE	TRUE: interrupt function block
usiCom	Main site index	USINT		1	Master station index 1: The first master station
uiDevice	Slave address	UINT		0	Used to access the EtherCAT slave address
usiChannel	aisle	USINT		1	No need to set, this is set automatically
uiIndex	index	UINT		0	Index of the object dictionary
usiSubIndex	Subindex	USINT		0	Sub-index of the object dictionary

udiTimeOut	Operation delay	UDINT		0	Set delay time, unit ms
pBuffer	Buffer zone	CAA.PVOID		0	Pointer to the buffer for writing data
szSize	Data size	CAA.SIZE		0	Transmission data size, unit byte
eMode	Mode selection	ETC_CO_MODE		0	Transmission mode: ETC_CO_AUTO; ETC_CO_EXPEDITED; ETC_CO_SEGMENTED (Ethercat does not support)

(2) Output variables

Output variable	Name	Type of data	Effective range	Content
xDone	Parameter setting succeeded	BOOL	TRUE, FALSE	TRUE: The execution of the function block is complete
xBusy	Function block is executing	BOOL	TRUE, FALSE	TRUE: The function block is running
xError	Error	BOOL	TRUE, FALSE	TRUE: The function block generates an exception and has stopped execution
eError	Error	ETC_CO_ERROR		Function block alarm
udiSdoAbort	SDO interrupt	U DINT		Abort code received from the device
szDataWritten	Data size	CAA.SIZE		Transmission data size, unit byte

(3) Conversion timing of output variables

Variable	When it becomes TRUE	When it becomes FALSE
xDone	✧ When reset is complete	✧ xExecute is TRUE, it is at the same time as the FALSE of xExecute ✧ xExecute is FALSE, after 1 cycle
xBusy	✧ xExecute rising edge	✧ xDone is TRUE ✧ xError is TRUE
xError	✧ When a fault occurs during the execution of a function block	✧ When the exception is resolved

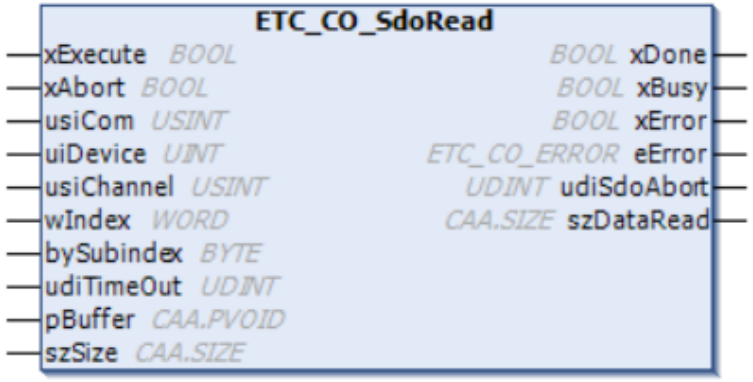
Key points

- xExecute is to start the function block. The rising edge can be triggered again during the execution of the function block. The input parameters of the function block will be reloaded every time the rising edge is executed, and the function block will be executed again.
- This function block is used to write Ether CAT slave parameters. To write parameters, you need to specify with uiIndex and usiSubIndex, and pBuffer needs to use pointers.
- unit size of szSize is byte, that is, szSize = 1 is 8 bits.
- transmission speed of ETC_CO_EXPEDITED in eMode is faster than ETC_CO_AUTO.
- uiDevice is shown in the figure below:

ETC_CO_SdoRead (FB)

Read the slave parameters.

Name	ETC_CO_SdoRead (Read slave parameters)		
Supported modes	CSP	CSV	CST
Graphical performance		ST performance _	




```
ETC_CO_ SdoRead (
xExecute :=,
xAbort : =,
usiCom : =,
uiDevice : =,
Channel : =,
wIndex : =,
bySubindex : =,
udiTimeOut : =,
pBuffer : =,
szSize : =,
xDone = >,
xBusy = >,
xError = >,
eError = >,
udiSdoAbort = >,
szDataRead => );
```

Variables

(1) Input variables

Input variable	Name	Type of data	Effective range	Initial value	Content

xExecute	Start up	BOOL	TRUE, FALSE	FALSE	TRUE: Enable function block
xAbort	Interrupt	BOOL	TRUE, FALSE	FALSE	TRUE: interrupt function block
usiCom	Main site index	USINT		1	Master station index 1: The first master station
uiDevice	Slave address	UINT		0	Used to access the EtherCAT slave address
usiChannel	aisle	USINT		1	No need to set, this is set automatically
uiIndex	index	UINT		0	Index of the object dictionary
usiSubIndex	Subindex	USINT		0	Sub-index of the object dictionary
udiTimeOut	Operation delay	UDINT		0	Set delay time, unit ms
pBuffer	Buffer zone	CAA.PVOID		0	Pointer to the buffer for writing data
szSize	Data size	CAA.SIZE		0	Transmission data size, unit byte

(2) Output variables

Output variable	Name	Type of data	Effective range	Content
xDone	Parameter setting succeeded	BOOL	TRUE, FALSE	TRUE: The execution of the function block is complete
xBusy	Function block is executing	BOOL	TRUE, FALSE	TRUE: The function block is running
xError	Error	BOOL	TRUE, FALSE	TRUE: The function block generates an exception and has stopped execution
eError	Error	ETC_CO_ERROR		Function block alarm
udiSdoAbort	S DO interrupt	U DINT		Abort code received from the device
szDataWritten	Data size	CAA.SIZE		Transmission data size, unit byte

(3) Conversion timing of output variables

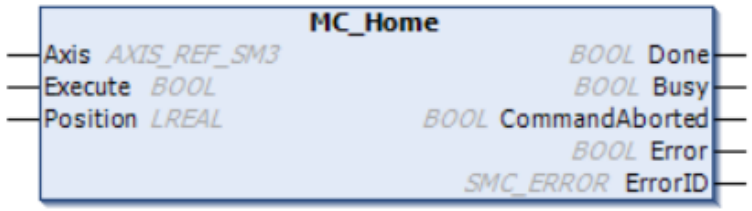
Variable	When it becomes TRUE	When it becomes FALSE
xDone	✧ When reset is complete	✧ xExecute is TRUE, it is at the same time as the FALSE of xExecute ✧ x When Execute is FALSE, after 1 cycle
xBusy	✧ xExecute rising edge	✧ xDone is TRUE ✧ xError is TRUE
xError	✧ When a fault occurs during the execution of a function block	✧ When the exception is resolved

Key points

- xExecute is to start the function block. The rising edge can be triggered again during the execution of the function block. The input parameters of the function block will be reloaded every time the rising edge is executed, and the function block will be executed again.
- This function block is used to read Ether CAT slave parameters. To read the parameters, you need to specify with uiIndex and usiSubIndex, and pBuffer needs to use pointers.
- unit size of szSize is byte, that is, szSize = 1 is 8 bits.
- uiDevice is shown in the figure below:

MC_Home (FB)

Control the axis to return the drive to the original.

Name	MC_Home (Axis back to the original)		
Supported modes	CSP		
Graphical performance		ST performance _	
		MC_ Home (Axis:=, Execute:=, Position:=, Done= >, Busy = >, CommandAborted = >, Error = >, ErrorID =>);	

Variables

(1) Input and output variables

Input and output variable	Name	Type of data	Content
Axis	Axis	AXIS_REF_SM3	Specify the axis, which is an instance of AXIS_REF_SM3

(2) Input variables

Input variable	Name	Type of data	Effective range	Initial value	Content
Execute	Start up	BOOL	TRUE, FALSE	FALSE	TRUE: Enable function block
Position	Origin setting position	LREAL	Negative number, positive number, "0"	0	Return to the original, the setting position of the origin

(3) Output variables

Output variable	Name	Type of data	Effective range	Content
Done	Finish	BOOL	TRUE, FALSE	TRUE: The execution of the function block is complete
Busy	Function block is executing	BOOL	TRUE, FALSE	TRUE: The function block is running
CommandAborted	Function block execution interrupt	BOOL	TRUE, FALSE	TRUE: The function block is aborted
Error	Error	BOOL	TRUE, FALSE	TRUE: The function block generates an exception and has stopped execution
ErrorID	Error code	SMC_ERROR	0	When an exception occurs, an error code is output

(4) Conversion timing of output variables

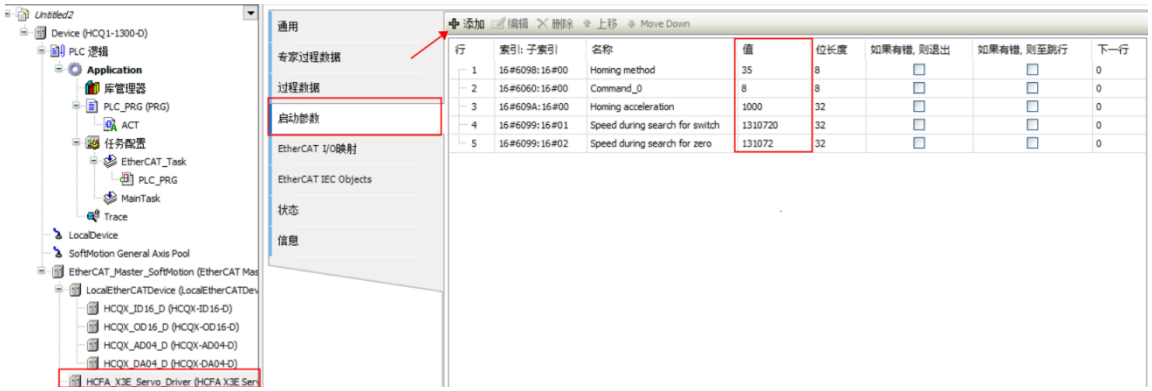
Variable	When it becomes TRUE	When it becomes FALSE
Done	✧ When the return is complete	✧ When Execute is TRUE, at the same time as the FALSE of Execute ✧ Execute is FALSE, after 1 cycle
Busy	✧ Rising edge of Execute	✧ When Done is TRUE ✧ When Error is TRUE ✧ Command Aborted becomes TRUE
CommandAborted	✧ Multi-start (interruption) of motion function blocks by other function blocks, when this function block is terminated ✧ When this function block is terminated due to an abnormality ✧ When an exception is occurring, this function block is started ✧ MC _Stop function block is being executed, this function block is started	✧ Execute is TRUE, it is at the same time as FALSE of Execute ✧ Execute is FALSE, after 1 cycle
Error	✧ When a fault occurs during the execution of a function block	✧ When the exception is resolved

Key points

- The rising edge of Execute is to start the function block. The rising edge can be triggered again during the execution of the function block. The input parameters of the function block will be reloaded each time the rising edge is executed, and the function block will be executed again.
- When the axis is in the STandstill state.
- When MC_Home is executed, other conventional motion function blocks cannot be executed, and MC_Stop can be used to stop and return.

- Before using, you need to configure the corresponding return parameters: 16 # 6098, 16 # 6099, 16 # 609A (For details on the return parameters, please refer to the drive manual).

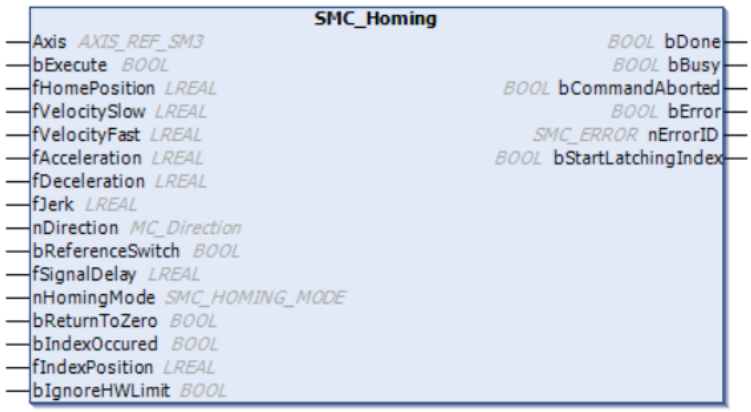
➢ **※Note: It is recommended to add the return to the original parameter in the startup parameter. Take HCFA X3EB drive as an example. The return parameter needs to pay attention to the servo P09.13 parameter. The third digit from the right of the P09.13 parameter can set the speed unit type, which is 1 o'clock. The speed unit is user command/S, when it is 0, the speed unit is revolutions per minute (rpm). The SDO parameters and values added in the figure below are for reference only, and the specific settings should be set according to the actual requirements of returning to the original.**



SMC_Homing (FB)

Control axis performs the controller return.

Name	SMC_Homing (Axis return)		
Supported modes	CSP		
Graphical performance		ST performance _	




```
SMC_ Homing (
Axis:=,
bExecute :=,
fHomePosition :=,
fVelocitySlow :=,
fVelocityFast :=,
fAcceleration :=,
fDeceleration:=,
fJerk:=,
nDirection:=,
bReferenceSwitch:=,
fSignalDelay:=,
nHomingMode:=,
bReturnToZero:=,
```

	bIndexOccured:=, fIndexPosition:=, bIgnoreHWLimit:=, bDone=>, bBusy=>, bCommandAborted = >, bError = >, nErrorID = >, bStartLatchingIndex =>);

Variables

(1) Input and output variables

Input and output variable	Name	Type of data	Content
Axis	Axis	AXIS_REF_SM3	Specify the axis, which is an instance of AXIS_REF_SM3

(2) Input variables

Input variable	Name	Type of data	Effective range	Initial value	Content
bExecute	Start up	BOOL	TRUE, FALSE	FALSE	TRUE: Enable function block
fHomePosition	Origin setting position	LREAL	Negative number, positive number, "0"	0	The set position of the origin after the return to original
fVelocitySlow	Slow speed	LREAL	A positive number	0	Return to the original slow running speed
fVelocityFast	Fast speed	LREAL	A positive number	0	Back to the original fast running speed
fAcceleration	Acceleration	LREAL	Positive number or "0"	0	Acceleration, the unit is [command unit/s^2]
fDeceleration	Deceleration	LREAL	Positive number or "0"	0	Deceleration, the unit is [command unit/s^2]
fJerk	Jerk	LREAL	Positive number or "0"	0	Specify jerk, the unit is [command unit/ S^3]
nDirection	Direction selection	MC_Direction		negative	Back to the original direction
bReferenceSwitch	Origin reference switch	BOOL	TRUE, FALSE	TRUE	The axis is not on the origin reference switch (bReferenceSwitch), the origin reference switch is TRUE. When the axis touches the origin reference switch, the origin reference switch is FALSE. After the axis leaves the reference, the origin reference switch is TRUE, and the return is completed.
fSignalDelay	Signal delay	LREAL		0	The transmission delay of the origin reference switch, the unit is S
nHomingMode	Back to the original	S MC _Homing_Mode		F AST_BSL	Return to the original mode selection, refer to S MC _Homing_Mode

	mode			OW_S_STOP	
bReturnToZero	Return to the original position	BOOL	TRUE, FALSE	FALSE	TRUE : After returning to the original, if fHomePosition is not set to 0, the axis will move to the position of 0 at the speed of fVelocityFast
bIndexOccured		BOOL	TRUE, FALSE	FALSE	TRUE : Take effect when the return mode is F AST_BSLOW_I_S_STIO or F AST_SLOW_I_S_STOP
fIndexPosition		LREAL		0	Position recorded at the mark pulse
bIgnoreHWLimit	Ignore hardware limit	BOOL	TRUE, FALSE	FALSE	TRUE: Ignore the hard limit function of the axis when returning to the original

(3) Output variable

Output variable	Name	Type of data	Effective range	Content
bDone	Finish	BOOL	TRUE, FALSE	TRUE: The execution of the function block is complete
bBusy	Function block is executing	BOOL	TRUE, FALSE	TRUE: The function block is running
bCommandAborted	Function block execution interrupt	BOOL	TRUE, FALSE	TRUE: The function block is aborted
bError	Error	BOOL	TRUE, FALSE	TRUE: The function block generates an exception and has stopped execution
nErrorID	error code	SMC_ERROR	0	When an exception occurs, an error code is output
bStartLatchingIndex	Start recording pulse	BOOL	TRUE, FALSE	Depends on the settings oFBIndexOccured and fIndexPosition.

(4) Conversion timing of output variables

Variable	When it becomes TRUE	When it becomes FALSE
bDone	✧ When the return is complete	✧ bExecute is TRUE at the same time as the FALSE of bExecute ✧ bExecute is FALSE, after 1 cycle
bBusy	✧ bExecute rising edge	✧ bDone is TRUE ✧ bError is TRUE ✧ CommandAborted becomes TRUE
bCommandAborted	✧ Multi-start (interruption) of motion function blocks by other function blocks, when this function block is terminated ✧ When this function block is terminated due to an abnormality ✧ When an exception is occurring, this function block is started ✧ MC _Stop function block is being executed, this function block is started	✧ bExecute is TRUE, it is at the same time as FALSE of bExecute ✧ bExecute is FALSE, after 1 cycle
bError	✧ When a fault occurs during the execution of a function block	✧ When the exception is resolved

Key points

- bExecute is to start the function block. The rising edge can be triggered again during the execution of the function block.

The input parameters of the function block will be reloaded each time the rising edge is executed, and the function block will be executed again.

- This function block is a return-to-origin method controlled by the controller, which is essentially different from MC _Home :

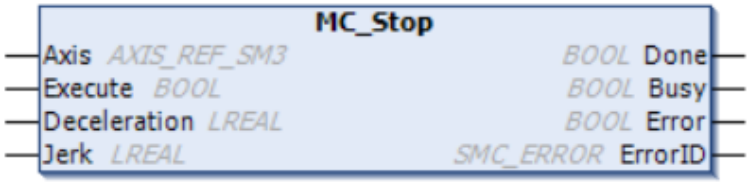
  MC _Home is the return to the original control of the drive, with higher accuracy, and will modify the actual position of the drive;

  SMC_Homing does not modify the actual position of the drive, but only sets the position through MC_SetPosition. The origin position may be lost after power failure.

- When the rising edge oFBExecute starts, SMC_Homing is executed, as long as the origin reference switch bReferenceSwitch is TRUE, the axis moves in the direction specified by nDirection at the speed of fVelocityFast. When the origin reference switch bReferenceSwitch is FALSE, the axis leaves the origin reference switch at a slow speed fVelocitySlow. When the origin reference switch bReferenceSwitch is TRUE again, it will act according to the selected homing mode nHomingMode, refer to SMC _Homing_Mode for details .

MC_Stop (FB)

Stop the movement of an axis and switch the axis to Stopping state.

Name	MC_S top (axis stop)		
Supported modes	CSP	CSV	
Graphical performance		ST performance _	
		MC_ Stop (Axis:=, Execute:=, Deceleration:=, Jerk:=, Done= >, Busy = >, Error = >, ErrorID =>);	

Variables

(1) Input and output variables

Input and output variable	Name	Type of data	Content

Axis	axis	AXIS_REF_SM3	Specify the axis, which is an instance of AXIS_REF_SM3

(2) Input variables

Input variable	Name	Type of data	Effective range	Initial value	Content
Execute	Start up	BOOL	TRUE, FALSE	FALSE	TRUE: Enable function block
Deceleration	Deceleration	LREAL	Positive number or "0"	0	Deceleration, the unit is [command unit/s^2]
Jerk	Jerk	LREAL	Positive number or "0"	0	Specify jerk, the unit is [command unit/S^3]

(3) Output variables

Output variable	Name	Type of data	Effective range	Content
Done	Finish	BOOL	TRUE, FALSE	TRUE: The execution of the function block is complete
Busy	Function block is executing	BOOL	TRUE, FALSE	TRUE: The function block is running
Error	Error	BOOL	TRUE, FALSE	TRUE: The function block generates an exception and has stopped execution
ErrorID	Error code	SMC_ERROR	0	When an exception occurs, an error code is output

Key points

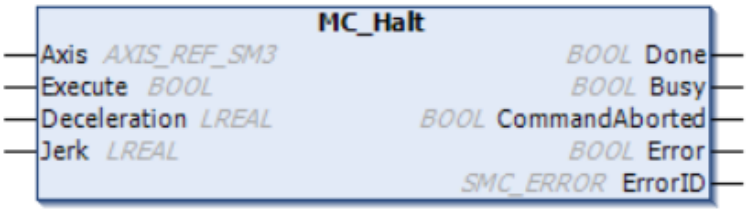
- The rising edge of Execute is to start the function block. The rising edge can be triggered again during the execution of the function block. The input parameters of the function block will be reloaded each time the rising edge is executed, and the function block will be executed again.
- This function block stops the movement of an axis under normal operation and switches the axis to Topping state. When the axis is in Stopping state, any movement function block is invalid.
- Even if MC _Stop stops the axis movement and Done has become TRUE, as long as Execute is not set to FALSE, the axis will still remain in the topping state.
- Multiple MC_Stops cannot be interrupted each other, so it is recommended to ensure that one axis in the program is controlled by only one MC_Stop. It is recommended to use MC_Halt for regular stop.

➢ ※ Note: Mc_Stop is not available in torque mode !

MC_Halt (FB)

Stop the movement of an axis.

Name	MC_Halt (axis stop)		
Supported modes	CSP	CSV	

Graphical performance	ST performance _
	MC_ Halt (Axis:=, Execute:=, Deceleration:=, Jerk:=, Done= >, Busy = >, CommandAborted = >, Error = >, ErrorID =>);

Variables

(1) Input and output variables

Input and output variable	Name	Type of data	Content
Axis	axis	AXIS_REF_SM3	Specify the axis, which is an instance of AXIS_REF_SM3

(2) Input variables

Input variable	Name	Type of data	Effective range	Initial value	Content
Execute	Start up	BOOL	TRUE, FALSE	FALSE	TRUE: Enable function block
Deceleration	Deceleration	LREAL	Positive number or "0"	0	Deceleration, the unit is [command unit/s^2]
Jerk	Jerk	LREAL	Positive number or "0"	0	Specify jerk, the unit is [command unit/S^3]

(3) Output variables

Output variable	Name	Type of data	Effective range	Content
Done	Finish	BOOL	TRUE, FALSE	TRUE: The execution of the function block is complete
Busy	Function block is executing	BOOL	TRUE, FALSE	TRUE: The function block is running
CommandAborted	Function block execution interrupt	BOOL	TRUE, FALSE	TRUE: The function block is aborted
Error	Error	BOOL	TRUE, FALSE	TRUE: The function block generates an exception and has stopped execution
ErrorID	Error code	SMC_ERROR	0	When an exception occurs, an error code is output

Key points

- The rising edge of Execute is to start the function block. The rising edge can be triggered again during the execution of

the function block. The input parameters of the function block will be reloaded each time the rising edge is executed, and the function block will be executed again.

- This function block is to stop the movement of an axis under normal operation, If other motion function blocks are triggered during the stop process, the execution of this function block will be interrupted.
- function block during execution is Discrete_motion, and the status is Standstill after the operation is completed.
- MC_Halt is different from MC_ Stop, MC_H alt is allowed to be interrupted.

MC _Jog (FB)

Control axis performs forward and reverse jogging.

Name	MC _J og (Axis Jog Control)		
Supported modes	CSP	CSV	
Graphical performance		ST performance _	
		MC_ Jog (Axis:=, JogForward :=, JogBackward :=, Velocity:=, Acceleration:=, Deceleration:=, Jerk:=, Busy = >, CommandAborted = >, Error = >, ErrorId =>);	

Variables

(1) Input and output variables

Input and output variable	Name	Type of data	Content
Axis	Axis	AXIS_REF_SM3	Specify the axis, which is an instance of AXIS_REF_SM3

(2) Input variables

Input variable	Name	Type of data	Effective range	Initial value	Content
JogForward	Forward jog	BOOL	TRUE, FALSE	FALSE	TRUE : The axis moves in the positive direction FALSE : The axis stops moving in the positive direction
JogBackward	Reverse jog	BOOL	TRUE, FALSE	FALSE	TRUE : the axis moves in the opposite direction

					FALSE : The axis stops moving in the opposite direction
Velocity	Target speed	LREAL	A positive number	0	Specify the target speed. The unit is [command unit/S]
Acceleration	Acceleration	LREAL	Positive number or "0"	0	Acceleration, the unit is [command unit/s^2]
Deceleration	Deceleration	LREAL	Positive number or "0"	0	Deceleration, the unit is [command unit/s^2]
Jerk	Jerk	LREAL	Positive number or "0"	0	Specify jerk, the unit is [command unit/S^3]

(3) Output variables

Output variable	Name	Type of data	Effective range	Content
Busy	Function block is executing	BOOL	TRUE, FALSE	TRUE: The function block is running
CommandAborted	Function block execution interrupt	BOOL	TRUE, FALSE	TRUE: The function block is aborted
Error	Error	BOOL	TRUE, FALSE	TRUE: The function block generates an exception and has stopped execution
ErrorID	Error code	SMC_ERROR	0	When an exception occurs, an error code is output

(4) Conversion timing of output variables

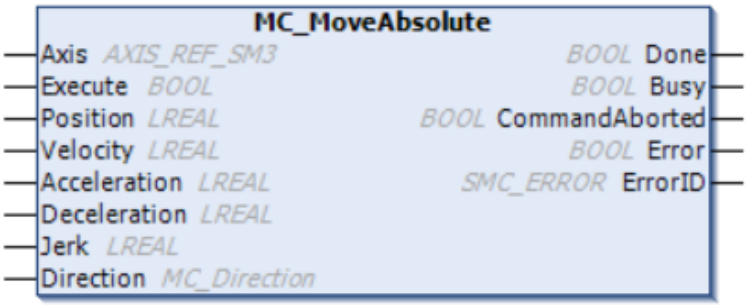
Variable	When it becomes TRUE	When it becomes FALSE
Busy	✧ JogForward or JogBackward is TRUE	✧ When Error is TRUE ✧ CommandAborted becomes TRUE
CommandAborted	✧ Multi-start (interruption) of motion function blocks by other function blocks, when this function block is terminated ✧ When this function block is terminated due to an abnormality ✧ When an exception is occurring, this function block is started ✧ MC _Stop function block is being executed, this function block is started	✧ Execute is TRUE, it is at the same time as FALSE of Execute ✧ Execute is FALSE, after 1 cycle
Error	✧ When a fault occurs during the execution of a function block	✧ When the exception is resolved

Key points

- Control axis is jog running, and the positive jog is controlled by JogForward. When set to TRUE, the axis will jog forward according to the set speed and acceleration. Reverse jogging is controlled by JogBackward. When set to TRUE, the axis will perform reverse jogging according to the set speed and acceleration.
- Jog operation, if JogForward or JogBackwar changes from TRUE to FALSE, the axis will immediately decelerate and stop at the set deceleration.
- During the jogging process, the axis is in the Continuous Motion state.
- If JogForward and JogBackward are set to TRUE at the same time, the axis will stop moving or not start moving and no error will be reported.

MC_MoveAbsolute (FB)

The control axis performs positioning motion at the absolute target position.

Name	MC_MoveAbsolute (axis absolute position control)		
Supported modes	CSP		
Graphical performance		ST performance _	
		MC_ MoveAbsolute (Axis:=, Execute:=, Position:=, Velocity:=, Acceleration:=, Deceleration:=, Jerk:=, Direction:=, Done= >, Busy = >, CommandAborted = >, Error = >,	

Variables

(1) Input and output variables

Input and output variable	Name	Type of data	Effective range	Content
Axis	Axis	AXIS_REF_SM3		Specify the axis, which is an instance of AXIS_REF_SM3

(2) Input variables

Input variable	Name	Type of data	Effective range	Initial value	Content
Execute	Start up	BOOL	TRUE, FALSE	FALSE	TRUE: Enable function block
Position	Target location	LREAL	Negative number, positive number, "0"	0	Specify the target position of absolute coordinates, the unit is [command unit]
Velocity	Target speed	LREAL	A positive number	0	Specify the target speed, the unit is [command unit/S]
Acceleration	Acceleration	LREAL	Positive number or "0"	0	Acceleration, the unit is [command unit/s^2]
Deceleration	Deceleration	LREAL	Positive number or "0"	0	Deceleration, the unit is [command unit/s^2]
Jerk	Jerk	LREAL	Positive number or	0	Specify jerk, the unit is [command unit/S^3]

			"0"		
Direction	Direction selection	MC_Direction	Fastest,current, positive, shortest, negative	shortest	Reference MC_Direction

(3) Output variable

Output variable	Name	Type of data	Effective range	Content
Done	Finish	BOOL	TRUE, FALSE	TRUE: The execution of the function block is complete
Busy	Function block is executing	BOOL	TRUE, FALSE	TRUE: The function block is running
CommandAborted	Function block execution interrupt	BOOL	TRUE, FALSE	TRUE: The function block is aborted
Error	Error	BOOL	TRUE, FALSE	TRUE: The function block generates an exception and has stopped execution
ErrorID	Error code	SMC_ERROR	0	When an exception occurs, an error code is output

(4) Conversion timing of output variables

Variable	When it becomes TRUE	When it becomes FALSE
Done	✧ When positioning is complete	✧ When Execute is TRUE, at the same time as the FALSE of Execute ✧ Execute is FALSE, after 1 cycle
Busy	✧ Rising edge of Execute	✧ When Done is TRUE ✧ When Error is TRUE ✧ Command Aborted becomes TRUE
CommandAborted	✧ Multi-start (interruption) of motion function blocks by other function blocks, when this function block is terminated ✧ When this function block is terminated due to an abnormality ✧ When an exception is occurring, this function block is started ✧ MC _Stop function block is being executed, this function block is started	✧ Execute is TRUE, it is at the same time as FALSE of Execute ✧ Execute is FALSE, after 1 cycle
Error	✧ When a fault occurs during the execution of a function block	✧ When the exception is resolved

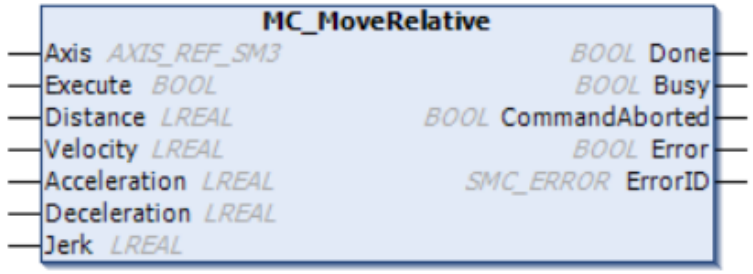
Key points

- This function block is an absolute positioning function block, and the Position data is the absolute position of the axis. The state of the axis when this function block is executed is Discrete Motion.
- The rising edge of Execute is to start the function block. The rising edge can be triggered again during the execution of the function block. The input parameters of the function block will be reloaded each time the rising edge is executed, and the function block will be executed again.
- Acceleration or Deceleration is zero, an error will be reported when the function block is started (Execute), and the state of the axis is Standstill.

- Direction (direction selection), refer to MC _Direction .

MC_MoveRelative (FB)

The control axis performs positioning motion relative to the target position.

Name	MC_MoveRelative (relative positioning)		
Supported modes	CSP		
Graphical performance		ST performance _	
		MC_ MoveRelative (Axis:=, Execute:=, Distance:=, Velocity:=, Acceleration:=, Deceleration:=, Jerk:=, Done= >, Busy = >, CommandAborted = >, Error = >,	

Variables

(1) Input and output variables

Input and output variables	Name	Type of data	Content
Axis	Axis	AXIS_REF_SM3	Specify the axis, which is an instance of AXIS_REF_SM3

(2) Input variables

Input variable	Name	Type of data	Effective range	Initial value	Content
Execute	Start up	BOOL	TRUE, FALSE	FALSE	TRUE: Enable function block
Distance	Relative target position	LREAL	Negative number, positive number, "0"	0	Specify the target position of the relative coordinate, the unit is [command unit]
Velocity	Target speed	LREAL	A positive number	0	Specify the target speed, the unit is [command unit/S]
Acceleration	Acceleration	LREAL	Positive number or "0"	0	Acceleration, the unit is [command unit/s^2]
Deceleration	Deceleration	LREAL	Positive number or "0"	0	Deceleration, the unit is [command unit/s^2]

Jerk	Jerk	LREAL	Positive number or "0"	0	Specify jerk, the unit is [command unit/S ^{3}]

(3) Output variable

Output variable	Name	Type of data	Effective range	Content
Done	Finish	BOOL	TRUE, FALSE	TRUE: The execution of the function block is complete
Busy	Function block is executing	BOOL	TRUE, FALSE	TRUE: The function block is running
CommandAborted	Function block execution interrupt	BOOL	TRUE, FALSE	TRUE: The function block is aborted
Error	Error	BOOL	TRUE, FALSE	TRUE: The function block generates an exception and has stopped execution
ErrorID	Error code	SMC_ERROR	0	When an exception occurs, an error code is output

(4) Conversion timing of output variables

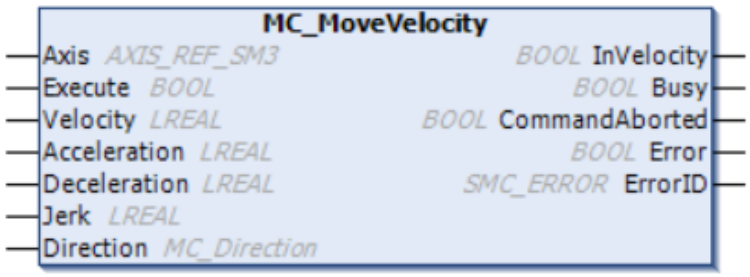
Variable	When it becomes TRUE	When it becomes FALSE
Done	✧ When positioning is complete	✧ When Execute is TRUE, at the same time as the FALSE of Execute ✧ Execute is FALSE, after 1 cycle
Busy	✧ Rising edge of Execute	✧ When Error is TRUE ✧ CommandAborted becomes TRUE
CommandAborted	✧ Multi-start (interruption) of motion function blocks by other function blocks, when this function block is terminated ✧ When this function block is terminated due to an abnormality ✧ When an exception is occurring, this function block is started ✧ MC _Stop function block is being executed, this function block is started	✧ Execute is TRUE, it is at the same time as FALSE of Execute ✧ Execute is FALSE, after 1 cycle
Error	✧ When a fault occurs during the execution of a function block	✧ When the exception is resolved

Key points

- This function block is a relative positioning function, D i stance is the relative distance that the axis will move.
- Discrete Motion state when the function block is executed.
- Execute is to start the function block, and the rising edge can be triggered again during the execution of the function block, and the input parameters of the function block will be reloaded every time the rising edge.
- Acceleration or Deceleration is zero, an error will be reported during the specified operation, and the state of the axis is Standstill.
- At the same time that Execute gets the rising edge action, Busy becomes TRUE. After reaching the target position, positioning is completed, Done becomes TRUE, and Busy becomes FALSE.
- When this function block is interrupted, CommandAborted becomes TRUE and Busy becomes FALSE.

MC _M ove Velocity (FB)

Control axis moves at a constant speed.

Name	MC _MoveVelocity (speed control)		
Supported modes	CSP	CSV	
Graphical performance		ST performance _	
		MC_ MoveVelocity (Axis:=, Execute:=, Velocity:=, Acceleration:=, Deceleration:=, Jerk:=, Direction:=, InVelocity = >, Busy = >, CommandAborted = >,	

Variables

(1) Input and output variables

Input and output variables	Name	Type of data	Content
Axis	Axis	AXIS_REF_SM3	Specify the axis, which is an instance of AXIS_REF_SM3

(2) Input variables

Input variable	Name	Type of data	Effective range	Initial value	Content
Execute	Start up	BOOL	TRUE, FALSE	FALSE	TRUE: Enable function block
Velocity	Target speed	LREAL	A positive number	0	Specify the target speed, the unit is [command unit/S]
Acceleration	Acceleration	LREAL	Positive number or "0"	0	Acceleration, the unit is [command unit/s^2]
Deceleration	Deceleration	LREAL	Positive number or "0"	0	Deceleration, the unit is [command unit/s^2]
Jerk	Jerk	LREAL	Positive number or "0"	0	Specify jerk, the unit is [command unit/S^3]
Direction	Direction selection	MC_Direction	fastest,current, positive, shortest, negative	shortest	Reference MC_Dire c tion

(3) Output variable

Output variable	Name	Type of data	Effective range	Content
InVelocity	Reached the set speed	BOOL	TRUE, FALSE	TRUE : Reached the set target speed
Busy	Function block is executing	BOOL	TRUE, FALSE	TRUE: The function block is running

CommandAborted	Function block execution interrupt	BOOL	TRUE, FALSE	TRUE: The function block is aborted
Error	Error	BOOL	TRUE, FALSE	TRUE: The function block generates an exception and has stopped execution
ErrorID	Error code	SMC_ERROR	0	When an exception occurs, an error code is output

(4) Conversion timing of output variables

variable	When it becomes TRUE	When it becomes FALSE
InVelocity	✧ The target speed reaches the set value	✧ The speed does not reach the set value of the target speed ✧ When Error is TRUE ✧ Command Aborted becomes TRUE
Busy	✧ Rising edge of Execute	✧ When Error is TRUE ✧ Command Aborted becomes TRUE
CommandAborted	✧ Multi-start (interruption) of motion function blocks by other function blocks, when this function block is terminated ✧ When this function block is terminated due to an abnormality ✧ When an exception is occurring, this function block is started ✧ MC _Stop function block is being executed, this function block is started	✧ Execute is TRUE, it is at the same time as FALSE of Execute ✧ Execute is FALSE, after 1 cycle
Error	✧ When a fault occurs during the execution of a function block	✧ When the exception is resolved

Key points

- This function block controls the axis to move continuously at the set speed.
- Axis is in Continuous Motion state when the function block is executed.
- Execute is to start the function block, and the rising edge can be triggered again during the execution of the function block, and the input parameters of the function block will be reloaded every time the rising edge.
- When the axis status is Standstill, the acceleration or deceleration of MC_MoveVelocity is 0, the function block will report an error, and the axis status is Standstill.
- the axis state is Continuous Motion, the acceleration or deceleration of MC_MoveVelocity is 0, the function block and the axis will report an error, and the axis state is Errorstop.
- Function block is executed, setting the Execute pin to FALSE will not stop the execution of the function block. The axis will continue to execute according to the set parameters. If you want to stop MC_MoveVelocity, you can execute MC_Halt to stop.

SMC_ Set Torque (FB)

The control axis outputs the set torque (torque mode is available).

Name	SMC _SetTorque (Torque setting)		
Supported modes	CST		
Graphical performance		ST performance _	
		SMC_ SetTorque (Axis:=, bEnable :=, fTorque :=, bBusy = >, bError = >, nErrorID =>);	

Variables

(1) Input and output variables

Input and output variables	Name	Type of data	Content
Axis	Axis	AXIS_REF_SM3	Specify the axis, which is an instance of AXIS_REF_SM3

(2) Input variables

Input variable	Name	Type of data	Effective range	Initial value	Content
bEnable	Effective	BOOL	TRUE, FALSE	FALSE	TRUE: Enable function block FALSE : Close the function block
fTorque	Set torque	LREAL		0	Unit: Nm

(3) Output variables

Output variable	Name	Type of data	Effective range	Content
bBusy	Function block is executing	BOOL	TRUE, FALSE	Becomes TRUE when the function block is received
bError	Error	BOOL	TRUE, FALSE	TRUE: The function block generates an exception and has stopped execution
nErrorID	error code	SMC_ERROR	0	When an exception occurs, an error code is output

(4) Conversion timing of output variables

variable	When it becomes TRUE	When it becomes FALSE
bBusy	✧ bEnable is TRUE	✧ When Done is TRUE

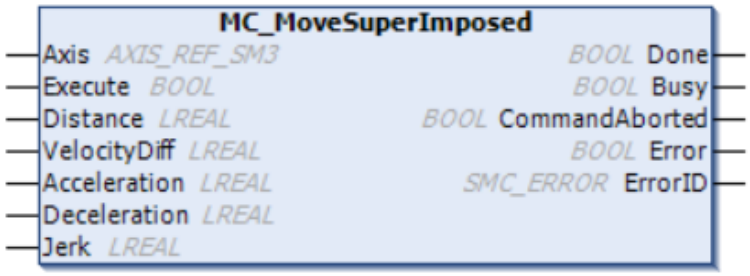
		✧ When Error is TRUE ✧ Command Aborted becomes TRUE
bError	✧ When a fault occurs during the execution of a function block	✧ When the exception is resolved

Key points

- This function block is to set the torque target value for the axis, and the torque target value will only take effect in the torque mode. Therefore, to use this function block, you must first ensure that the axis has been switched to the torque mode.
- Axis control mode can be switched through the SMC_SetControllerMode function block.
- This function block assigns 6071h in the object dictionary and requires 6071h in RXPDO.

MC_MoveSuperImposed (FB)

Superimpose position and speed motion function.
The position and speed are superimposed on the basis of the original control function block of the axis, without interrupting or affecting the original control function block of the axis movement.

Name	M c_MoveSuperImposed (Superimposed relative motion)		
Supported modes	CSP		
Graphical performance		ST performance _	
		MC_ MoveSuperImposed (Axis:=, Execute:=, Abort:=, Distance:=, VelocityDiff :=, Acceleration:=, Deceleration:=, Jerk:=, Done= >, Busy = >, CommandAborted = >, Error = >, ErrorID =>);	

Variables

(1) Input and output variables

Input and output variables	Name	Type of data	Content

Axis	axis	AXIS_REF_SM3	Specify the axis, which is an instance of AXIS_REF_SM3

(2) Input variables

Input variable	Name	Type of data	Effective range	Initial value	Content
Execute	Start up	BOOL	TRUE, FALSE	FALSE	TRUE: Enable function block
Abort	Interrupt	BOOL	TRUE, FALSE	FALSE	Interrupt when input a rise
Distance	Relative target position	LREAL	Negative number, positive number, "0"	0	Specify the target position of the relative coordinate, the unit is [command unit]
VelocityDiff	Superimposition speed	LREAL	A positive number	0	Specify superimposing speed, the unit is [command unit/ S]
Acceleration	Acceleration	LREAL	Positive number or "0"	0	Acceleration, the unit is [command unit/s^2]
Deceleration	Deceleration	LREAL	Positive number or "0"	0	Deceleration, the unit is [command unit/s^2]
Jerk	Jerk	LREAL	Positive number or "0"	0	Specify jerk, the unit is [command unit/S^3]

(3) Output variables

Output variable	Name	Type of data	Effective range	Content
Done	Finish	BOOL	TRUE, FALSE	TRUE: The execution of the function block is complete
Busy	Function block is executing	BOOL	TRUE, FALSE	TRUE: The function block is running
CommandAborted	Function block execution interrupt	BOOL	TRUE, FALSE	TRUE: The function block is aborted
Error	Error	BOOL	TRUE, FALSE	TRUE: The function block generates an exception and has stopped execution
ErrorID	Error code	SMC_ERROR	0	When an exception occurs, an error code is output

(4) Conversion timing of output variables

variable	When it becomes TRUE	When it becomes FALSE
Done	✧ When positioning is complete	✧ When Execute is TRUE, at the same time as the FALSE of Execute ✧ Execute is FALSE, after 1 cycle
Busy	✧ Rising edge of Execute	✧ When Done is TRUE ✧ When Error is TRUE ✧ Command Aborted becomes TRUE
CommandAborted	✧ Multi-start (interruption) of motion function blocks by other function blocks, when this function block is terminated ✧ When this function block is terminated due to an abnormality ✧ When an exception is occurring, this function block is started ✧ MC _Stop function block is being executed, this function block is	✧ Execute is TRUE, it is at the same time as FALSE of Execute ✧ Execute is FALSE, after 1 cycle

	started	
Error	✧ When a fault occurs during the execution of a function block	✧ When the exception is resolved

Key points

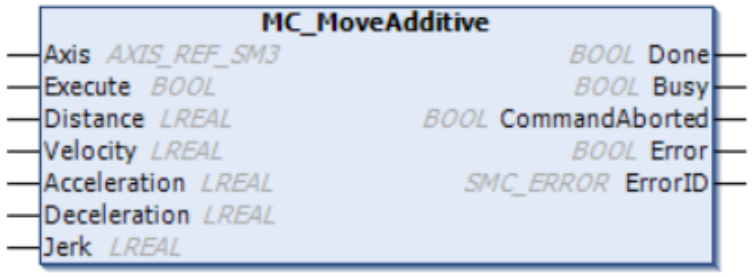
- This function block is the superimposed position and speed function block, D i stanc e is the distance superimposed on the basis of the executing function block, and VelocityDiff is the speed superimposed on the basis of the executing function block.
- Execute is to start the function block, and the rising edge can be triggered again during the execution of the function block, and the input parameters of the function block will be reloaded every time the rising edge. Under the premise that the last superimposition has not been completed, after the new superimposition data is loadeDINTo the function block, the final axis position is the target position of the positioning function block + the superimposition distance of the last superimposition.
- If this function block is executed alone, it is equivalent to MC_MoveRelative.
- When using this function block, you must pay attention to the order in which the function blocks are placed. This function block must be placed after the motion control function block that originally controlled the axis. If the superimposed axis serves as the master axis of other axes at the same time, before calling the synchronous motion control function block, it must be ensured that the MC_MoveSuperImposed of the master axis has been scanned and executed.
- The difference with MC_MoveAdditive :
  ① This function block does not cover the speed and other parameters of the original function block, but only superimposes a relative positioning function on the basis of the movement of the original function block. MC _MoveAdditive will interrupt the original function block, add the set position on the basis of the original function block, and then perform positioning movement at the speed set in the MoveAdditive pin.
  ② This function block is triggered repeatedly, and the superimposed position will not accumulate. The overlapping position of MC _MoveAdditive multiple triggers will accumulate, such as 1 00 each time. If MC _MoveAdditive is repeatedly triggered multiple times within several synchronization cycles, then the final overlapping position is: 100 *N (N=number of repeated triggers).
  ③ MC_MoveAdditive function block can be placed in front of the motion control function block that originally controlled the axis.

MC _MoveAdditive (FB)

Superimposed position movement function.

Perform position superposition on the basis of the original control function block of the axis, and cover the original speed, acceleration and other parameters.

Name	MC _M ove Additive (addition of absolute motion power)		
Supported modes	CSP		
Graphical performance		ST performance _	




```
MC_ MoveAdditive (
Axis:=,
Execute:=,
Distance:=,
Velocity:=,
Acceleration:=,
Deceleration:=,
Jerk:=,
Done= >,
Busy = >,
CommandAborted = >,
Error = >,
ErrorID => );
```

Variables

(1) Input and output variables

Input and output variables	Name	Type of data	Content
Axis	Axis	AXIS_REF_SM3	Specify the axis, which is an instance of AXIS_REF_SM3

(2) Input variables

Input variable	Name	Type of data	Effective range	Initial value	Content
Execute	Start up	BOOL	TRUE, FALSE	FALSE	TRUE: Enable function block
Distance	Absolute target position	LREAL	Negative number, positive number, "0"	0	Specify the target position of the relative coordinate, the unit is [command unit]
Velocity	Superimposition speed	LREAL	A positive number	0	Specify superimposing speed, the unit is [command unit/S]
Acceleration	Acceleration	LREAL	Positive number or "0"	0	Acceleration, the unit is [command unit/s^2]
Deceleration	Deceleration	LREAL	Positive number or "0"	0	Deceleration, the unit is [command unit/s^2]
Jerk	Jerk	LREAL	Positive number or "0"	0	Specify jerk, the unit is [command unit/S^3]

(3) Output variable

Output variable	Name	Type of data	Effective range	Content
Done	Finish	BOOL	TRUE, FALSE	TRUE: The execution of the function block is complete
Busy	Function block is executing	BOOL	TRUE, FALSE	TRUE: The function block is running

CommandAborted	Function block execution interrupt	BOOL	TRUE, FALSE	TRUE: The function block is aborted
Error	Error	BOOL	TRUE, FALSE	TRUE: The function block generates an exception and has stopped execution
ErrorID	Error code	SMC_ERROR	0	When an exception occurs, the error code is output

(4) Conversion timing of output variables

variable	When it becomes TRUE	When it becomes FALSE
Done	✧ When positioning is complete	✧ When Execute is TRUE, at the same time as FALSE of Execute ✧ Execute is FALSE, after 1 cycle
Busy	✧ Rising edge of Execute	✧ When Done is TRUE ✧ When Error is TRUE ✧ Command Aborted becomes TRUE
CommandAborted	✧ Multi-start (interruption) of motion function blocks by other function blocks, when this function block is terminated ✧ When this function block is terminated due to an abnormality ✧ When an exception is occurring, this function block is started ✧ MC _Stop function block is being executed, this function block is started	✧ Execute is TRUE, it is at the same time as FALSE of Execute ✧ Execute is FALSE, after 1 cycle
Error	✧ When a fault occurs during the execution of a function block	✧ When the exception is resolved

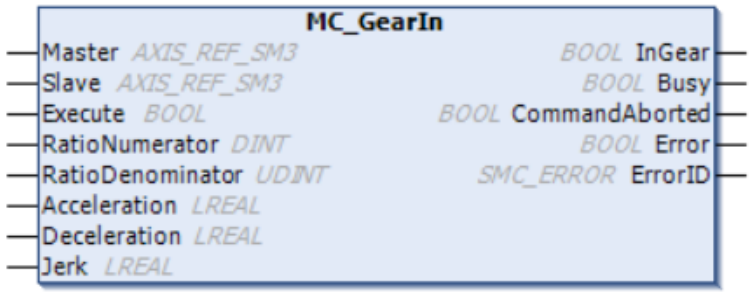
Key points

- This function block is the superimposed position, D i stance is the relative distance superimposed on the original position function block, and Velocity is the operating speed of the axis after the execution of this function block.
- MC _MoveAdditive interrupts other function blocks that can be interrupted when superimposed.
- Execute is to start the function block. The rising edge can be re-triggered during the execution of the function block. Each rising edge will reload the input parameters of the function block and superimpose a new position based on the position of the last trigger superimposition.
- If this function block is executed alone, it is equivalent to MC_MoveRelative.
- with MC _MoveSuperImposed : see MC _MoveSuperImposed .

MC GearIn (FB)

Electronic gear coupling function.
Control the slave axis to follow the master axis to run synchronously according to the set speed ratio.

Name	MC _GearIn (electronic gear coupling)		
Supported modes	CSP	CSV	
Graphical performance		ST performance _	




```
MC_ GearIn (
Master:=,
Slave:=,
Execute:=,
RatioNumerator :=,
RatioDenominator :=,
Acceleration:=,
Deceleration:=,
Jerk:=,
InGear = >,
Busy = >,
CommandAborted = >,
Error = >,
ErrorID => );
```

Variables

(1) Input and output variables

Input and output variables	Name	Type of data	Content
Master	Master axis	AXIS_REF_SM3	Specify the master axis
Slave	Slave axis	AXIS_REF_SM3	Specify slave axis

(2) Input variable

Input variable	Name	Type of data	Effective range	Initial	Content

				value	
Execute	Start up	BOOL	TRUE, FALSE	FALSE	TRUE: Enable function block
RatioNumerator	Gear ratio numerator	DINT	Negative, positive	1	Gear ratio numerator
RatioDenominator	Gear ratio denominator	UDINT	A positive number	1	Gear ratio denominator
Acceleration	Acceleration	LREAL	Positive number or "0"	0	Acceleration, the unit is [command unit/s^2]
Deceleration	Deceleration	LREAL	Positive number or "0"	0	Deceleration, the unit is [command unit/s^2]
Jerk	Jerk	LREAL	Positive number or "0"	0	Specify jerk, the unit is [command unit/S^3]

(3) Output variables

Output variable	Name	Type of data	Effective range	Content
InGear	Reach gear ratio	BOOL	TRUE, FALSE	TRUE: The slave axis is synchronized with the master axis.
Busy	Function block is executing	BOOL	TRUE, FALSE	TRUE: The function block is running
CommandAborted	Function block execution interrupt	BOOL	TRUE, FALSE	TRUE: The function block is aborted
Error	Error	BOOL	TRUE, FALSE	TRUE: The function block generates an exception and has stopped execution
ErrorID	Error code	SMC_ERROR	0	When an exception occurs, an error code is output

(4) Conversion timing of output variables

Variable	When it becomes TRUE	When it becomes FALSE
InGear	✧ The slave axis is synchronized with the master axis	✧ Execute MC_GearOut or other motion control functions to interrupt MC_GearIn ✧ Execute is FALSE, after 1 cycle
Busy	✧ Rising edge of Execute	✧ When Error is TRUE ✧ Command Aborted becomes TRUE
CommandAborted	✧ Multi-start (interruption) of motion function blocks by other function blocks, when this function block is terminated ✧ When this function block is terminated due to an abnormality ✧ When an exception is occurring, this function block is started ✧ MC_Stop function block is being executed, this function block is started	✧ Execute is TRUE, it is at the same time as FALSE of Execute ✧ Execute is FALSE, after 1 cycle
Error	✧ When a fault occurs during the execution of a function block	✧ When the exception is resolved

Key points

- This function block is the electronic gear coupling function. RatioNumerator and RatioDenominator are the numerator and denominator of the electronic gear ratio respectively. The speed synchronization relationship can refer to the following formula:

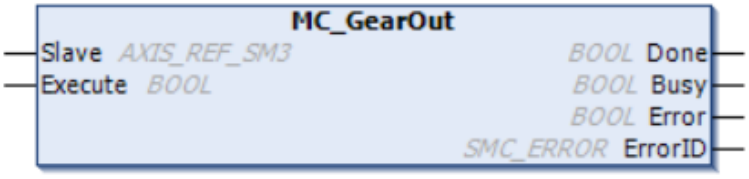
$$\text{Slave speed} = \frac{\text{RatioNumerator}}{\text{RatioDenominator}} * \text{Master axis speed}$$

- Execute is to start the function block. The rising edge can be triggered again during the execution of the function block. The input parameters of the function block will be reloaded each time the rising edge is executed, and the function block will be executed again.
- Function block is executed, the slave axis is in the Synchronized Motion state.
- This function block is only for the slave axis, and the master axis will not have any influence due to the slave axis coupling with it.
- The Acceleration and Deceleration parameters of the function block only take effect when the function block has just executed the slave axis following the Master axis speed. Once the slave axis and the Master axis speed are synchronized, the InGear pin is TRUE, and the function block will not perform acceleration/deceleration calculations., The speed of the master axis is directly converted to the slave axis after the gear ratio is converted.
- Modifying the RatioNumerator and RatioDenominator parameters during the execution of the function block will not take effect.

MC_GearOut (FB)

Electronic gear decoupling function.

This function will decouple the axis from the synchronization state with the master axis.

Name	Mc _GearOut (electronic gear decoupling)		
Supported modes	CSP	CSV	
Graphical performance		ST performance _	
		MC_GearOut (Slave:=, Execute:=, Done= >, Busy = >, Error = >, ErrorID =>);	

Variables

(1) Input and output variables

Input and output variables	Name	Type of data	Content
Slave	Slave axis	AXIS_REF_SM3	Specify slave axis

(2) Input variables

Input variable	Name	Type of data	Effective range	Initial value	Content
Execute	Start up	BOOL	TRUE, FALSE	FALSE	TRUE: Enable function block

(3) Output variables

Output variable	Name	Type of data	Effective range	Content
Done	Finish	BOOL	TRUE, FALSE	TRUE: The execution of the function block is complete
Busy	Function block is executing	BOOL	TRUE, FALSE	TRUE: The function block is running
Error	Error	BOOL	TRUE, FALSE	TRUE: The function block generates an exception and has stopped execution
ErrorID	Error code	SMC_ERROR	0	When an exception occurs, an error code is output

(4) Conversion timing of output variables

Variable	When it becomes TRUE	When it becomes FALSE
Done	✧ When decoupling is complete	✧ When Execute is TRUE, at the same time as FALSE of Execute ✧ Execute is FALSE, after 1 cycle
Busy	✧ Rising edge of Execute	✧ When Done is TRUE ✧ When Error is TRUE ✧ Command Aborted becomes TRUE
Error	✧ When a fault occurs during the execution of a function block	✧ When the exception is resolved

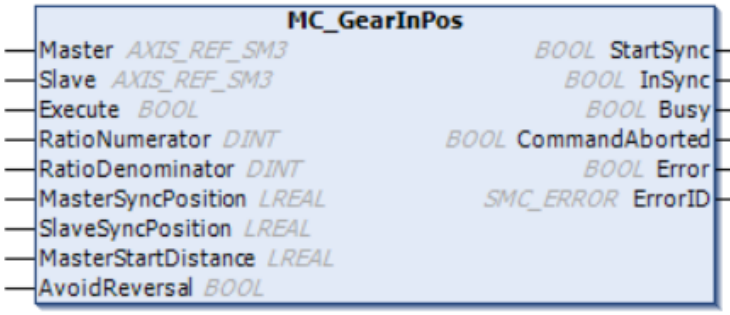
Key points

- The rising edge of Execute is to start the function block. The rising edge can be triggered again during the execution of the function block. Each rising edge will reload the input parameters of the function block and execute the function block again.
- Gear decoupling function block, the function is to separate from the slave axis and the master axis, it is valid for MC_GearIn and MC_GearInPos.
- This function block is the electronic gear coupling release function, The slave axis is disconnected from the synchronization state with the master axis, and the speed synchronization with the master axis is no longer performed.
- Function block is completed, the slave axis will be in the Continuous Motion state, and the slave axis will continue to run at the speed at which it was decoupled until it is taken over by other motion control function blocks. Therefore, it is recommended that if you want to stop the slave axis after decoupling, you can call MC _Halt to stop the slave axis after MC_GearOut is executed successfully, or directly use MC_Halt to interrupt the gear coupling.

MC_GearInPos (FB)

Slave axis position and master axis starting synchronization distance by setting the specified master axis position.

Name	MC _GearInPos (Designated position coupling electronic gear)		
Supported modes	CSP	CSV	
Graphical performance		ST performance _	




```
MC_ GearInPos (
Master:=,
Slave:=,
Execute:=,
RatioNumerator :=,
RatioDenominator :=,
MasterSyncPosition :=,
SlaveSyncPosition :=,
MasterStartDistance :=,
AvoidReversal :=,
StartSync = >,
InSync = >,
Busy = >,
CommandAborted = >,
Error = >,
ErrorID => );
```

Variables

(1) Input and output variables

Input and output variables	Name	Type of data	Content
Master	Master axis	AXIS_REF_SM3	Specify the master axis
Slave	Slave axis	AXIS_REF_SM3	Specify slave axis

(2) Input variable

Input variable	Name	Type of data	Effective range	Initial value	Content
Execute	Start up	BOOL	TRUE, FALSE	FALSE	TRUE: Enable function block
RatioNumerator	Gear ratio numerator	DINT	Negative, positive	1	Gear ratio numerator
RatioDenominator	Gear ratio denominator	U DINT	A positive number	1	Gear ratio denominator
MasterSyncPosition	Master axis synchronization position	LREAL			Master axis position when master-slave gear ratio is coupled
SlaveSyncPosition	Slave axis synchronization	LREAL			position when master-slave gear ratio is coupled

	position				
MasterStartDistance	Execute master axis synchronous position	LREAL			At this distance, the slave axis enters synchronization, if MasterStartDistance is 0, the slave axis starts immediately
AvoidReversal	No reversal	BOOL	TRUE, FALSE	FALSE	FALSE : If the physical position of the slave axis is advanced, it will be reversed TRUE : Only used under modal axis. If the reversal cannot be avoided, the axis will stop by Error

(3) Output variables

Output variable	Name	Type of data	Effective range	Content
StartSync	Start the coupling process	BOOL	TRUE, FALSE	When the electronic gear coupling starts, it becomes TRUE
InSync	Coupling	BOOL	TRUE, FALSE	When the electronic gear coupling is completed, it becomes TRUE
Busy	Function block is executing	BOOL	TRUE, FALSE	TRUE: The function block is running
CommandAborted	Function block execution interrupt	BOOL	TRUE, FALSE	TRUE: The function block is aborted
Error	Error	BOOL	TRUE, FALSE	TRUE: The function block generates an exception and has stopped execution
ErrorID	Error code	SMC_ERROR	0	When an exception occurs, an error code is output

(4) Conversion timing of output variables

Variable	When it becomes TRUE	When it becomes FALSE
StartSync	✧ Electronic gears start to couple	✧ When Error is TRUE ✧ Command Aborted becomes TRUE
InSync	✧ When the electronic gear is coupled	✧ When Error is TRUE ✧ Command Aborted becomes TRUE
Busy	✧ Rising edge of Execute	✧ When Error is TRUE ✧ Command Aborted becomes TRUE
CommandAborted	✧ Multi-start (interruption) of motion function blocks by other function blocks, when this function block is terminated ✧ When this function block is terminated due to an abnormality ✧ When an exception is occurring, this function block is started ✧ MC _Stop function block is being executed, this function block is started	✧ Execute is TRUE, it is at the same time as FALSE of Execute ✧ Execute is FALSE, after 1 cycle
Error	✧ When a fault occurs during the execution of a function block	✧ When the exception is resolved

Key points

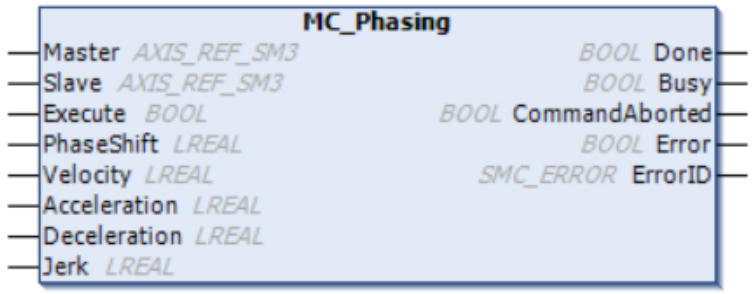
- The rising edge of Execute is to start the function block. The rising edge can be triggered again during the execution of the function block. The input parameters of the function block will be reloaded each time the rising edge is executed,

and the function block will be executed again. After the function block starts to operate, the slave axis uses the speed of the master axis multiplied by the gear ratio as the target speed, and performs acceleration and deceleration.

- Slave axis enters synchronization is equal to MasterSyncPosition-MasterStartDistance.

MC_Phasing (FB)

Modify the phase offset between the master axis and the slave axis

Name	MC_Phasing (Phase shift of master and slave axis)		
Supported modes	CSP		
Graphical performance		ST performance _	
		MC_ Phasing (Master:=, Slave:=, Execute:=, PhaseShift :=, Velocity:=, Acceleration:=, Deceleration:=, Jerk:=, Done= >, Busy = >, CommandAborted = >, Error = >, ErrorID =>);	

Variables

(1) Input and output variables

Input and output variables	Name	Type of data	Content
Master	Master axis	AXIS_REF_SM3	Specify the master axis
Slave	Slave axis	AXIS_REF_SM3	Map to slave axis

(2) Input variables

Input variable	Name	Type of data	Effective range	Initial value	Content
Execute	Start up	BOOL	TRUE, FALSE	FALSE	TRUE: Enable function block
PhaseShift	Phase offset	LREAL		0	Offset between master and slave axis
Velocity	Target speed	LREAL	A positive number	0	Specify the target speed. The unit is [command unit/S]
Acceleration	Acceleration	LREAL	Positive number or "0"	0	Acceleration, the unit is [command unit/s^2]
Deceleration	Deceleration	LREAL	Positive number or "0"	0	Deceleration, the unit is [command unit/s^2]
Jerk	Jerk	LREAL	Positive number or "0"	0	Specify jerk. The unit is [command unit/S^3]

(3) Output variables

Output variable	Name	Type of data	Effective range	Content
Done	Finish	BOOL	TRUE, FALSE	TRUE: The execution of the function block is complete
Busy	Function block is executing	BOOL	TRUE, FALSE	TRUE: The function block is running
CommandAborted	Function block execution interrupt	BOOL	TRUE, FALSE	TRUE: The function block is aborted
Error	Error	BOOL	TRUE, FALSE	TRUE: The function block generates an exception and has stopped execution
ErrorID	Error code	SMC_ERROR		When an exception occurs, an error code is output

(4) Conversion timing of output variables

variable	When it becomes TRUE	When it becomes FALSE
Done	✧ When finished	✧ When Execute is TRUE, at the same time as FALSE of Execute ✧ Execute is FALSE, after 1 cycle
Busy	✧ Rising edge of Execute	✧ When Done is TRUE ✧ When Error is TRUE ✧ Command Aborted becomes TRUE
CommandAborted	✧ Multi-start (interruption) of motion function blocks by other function blocks, when this function block is terminated ✧ When this function block is terminated due to an abnormality ✧ When an exception is occurring, this function block is started MC _Stop function block is being executed, this function block is started	✧ Execute is TRUE, it is at the same time as FALSE of Execute ✧ Execute is FALSE, after 1 cycle
Error	✧ When a fault occurs during the execution of a function block	✧ When the exception is resolved

Key points

- This function block adjusts the phase offset of the slave axis relative to the master axis. The slave axis will be adjusted according to the phase offset set by PhaseShift and the current actual phase offset, according to parameters such as Velocity, Acceleration, Deceleration, etc.
- PhaseShift sets the phase difference between the master axis and the slave axis. If PhaseShift is a positive number, it means that the phase of the master axis is ahead of the phase of the slave axis.
- The phase adjustment will not interrupt the original synchronous movement. After the phase adjustment is completed, if the slave axis is not decoupled, the phase difference between the master and slave axis will always be maintained.
- Function block is executed, the slave axis is in the Synchronized Motion state. This function block is only for the slave axis, and the master axis will not have any influence due to the slave axis coupling with it.
- Execute is to start the function block. The rising edge can be triggered again during the execution of the function block.

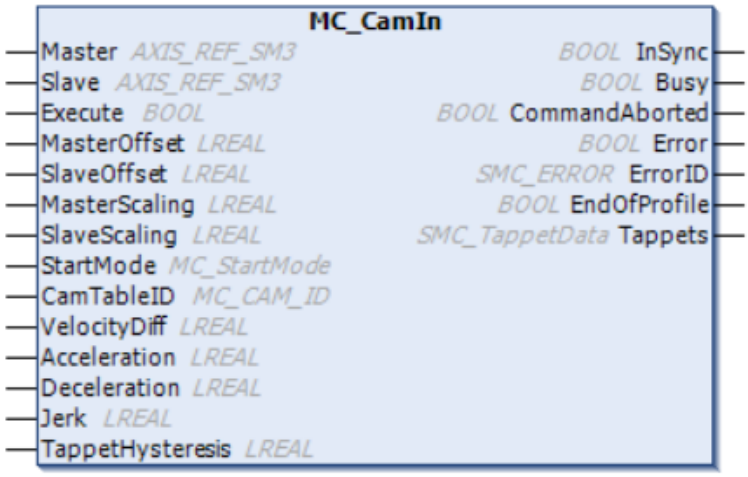
The input parameters of the function block will be reloaded each time the rising edge is executed, and the function block will be executed again. If the old phase deviation is not in place, the new phase deviation shall prevail.

MC_CamIn (FB)

Electronic cam coupling function.

Control the movement of the slave axis according to the position relationship specified by the cam table and the position of the master axis.

Name	MC_CamIn (master-slave coupling)		
Supported modes	CSP		
Graphical performance		ST performance _	




```
MC_ CamIn (
Master:=,
Slave:=,
Execute:=,
MasterOffset :=,
SlaveOffset :=,
MasterScaling :=,
SlaveScaling :=,
StartMode :=,
CamTableID :=,
VelocityDiff :=,
Acceleration:=,
Deceleration:=,
Jerk:=,
TappetHysteresis :=,
InSync = >,
Busy = >,
CommandAborted = >,
Error = >,
ErrorID = >,
EndOfProfile = >,
Tappets=> );
```

Variables

(1) Input and output variables

Input and output variables	Name	Type of data	Content
Master	Master axis	AXIS_REF_SM3	Map to master axis
Slave	Slave axis	AXIS_REF_SM3	Map to slave axis

(2) Input variables

Input variable	Name	Type of data	Effective range	Initial value	Content
Execute	Start up	BOOL	TRUE, FALSE	FALSE	TRUE: Enable function block
MasterOffset	Master axis	LREAL	Negative	0	Move the phase of the master axis by the specified offset

	position offset		number, positive number, "0"		value
SlaveOffset	Offset from axis position	LREAL	Negative number, positive number, "0"	0	Move the phase of the slave axis by the specified offset value
MasterScaling	Master axis magnification ratio	LREAL	> 0.0	1	Specify the master axis ratio
SlaveScaling	Slave magnification ratio	LREAL	> 0.0	1	Specify the slave axis scale
StartMode	Slave trigger mode	MC _StarMode		absolute	Reference MC _StarMode
CamTableID	Effective C am_ID	MC_CAM_ID			Define the use of the cam table, and use it in conjunction with the output point C amTableID of MC _CamTableSelect
VelocityDiff	Superimposition speed	LREAL	A positive number	0	Specify superimposing speed, the unit is [command unit/S]
Acceleration	Acceleration	LREAL	Positive number or "0"	0	Acceleration, the unit is [command unit/s^2]
Deceleration	Deceleration	LREAL	Positive number or "0"	0	Deceleration, the unit is [command unit/s^2]
Jerk	Jerk	LREAL	Positive number or "0"	0	Specify jerk, the unit is [command unit/S^3]
TappetHysteresis	Tappet lag	LREAL			Damping coefficient of tappet

(3) Output variables

Output variable	Name	Type of data	Effective range	Content
Busy	Function block is executing	BOOL	TRUE, FALSE	TRUE: The function block is running
Error	Error	BOOL	TRUE, FALSE	TRUE: The function block generates an exception and has stopped execution
ErrorID	Error code	SMC_ERROR		When an exception occurs, an error code is output
EndOfProfile	Curve complete	BOOL	TRUE, FALSE	TRUE: Cam execution ends. If the Periodic parameter of Mc_CamTableSelect selects TRUE (period), this pin will be TRUE in the period when the cam Master axis crosses the end point of the cam table If the Periodic parameter of Mc_CamTableSelect is FALSE (non-periodic), the cam will remain TRUE after the completion of the cam operation
Tappets	Tappet	SMC_TAPPETDATA		Associated with a cam tappet, which can be read by MC_Ge tTappetValue

(4) Conversion timing of output variables

Variable	When it becomes TRUE	When it becomes FALSE
Busy	✧ Rising edge of Execute	✧ When Done is TRUE ✧ When Error is TRUE ✧ Command Aborted becomes TRUE
Error	✧ When a fault occurs during the execution of a function block	✧ When the exception is resolved

Key points

- The rising edge of Execute is to start the function block. The rising edge can be triggered again during the execution of the function block. Each rising edge will reload the input parameters of the function block and execute the function block again.
- This function block is to couple the position relationship between the master and slave axes in the cam table. Before using this function block, you need to draw the cam curve and execute MC_CamTableSelect to select the cam table, and then execute MC_CamIn for cam coupling.
- If you want to change the cam curve relationship, you need to re-execute the MC_ CamTableSelect function block to select a new cam table at an appropriate time, and then execute MC_CamIn again.
- After the cam is coupled, the master axis and the slave axis have a coordinate system position X (corresponding to the master axis position in the cam table) and Y (slave axis position corresponding to the master axis position in the cam table) in the cam relationship, corresponding to the position in the cam table. These two positions are obtained by a series of calculation transformations of the actual Master axis and slave axis positions, and the change relationship can be abbreviated as follows:

X = MasterScaling * MasterPos + Ma sterOffset;

Y = CAM(X);

SlavePos = SlaveScaling * CAM(X) + SlaveOffset;

MasterPos : the actual position of the master axis, SlavePos : the target position of the slave axis, CAM(X) : calculate the value of Y according to the relationship between the value of X and the cam.

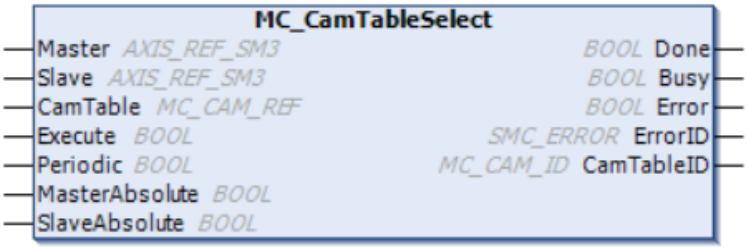
StartMode supports settings: Absolute and Relative modes. This mode setting is mainly for the slave axis. There are also two corresponding pins in MC_CamTableSelect to specify the absolute and relative relationship between the master axis and the slave axis. The specific coupling relationship is as follows:

MC_CamTableSelect.MasterAbsolute	final coordinate relationship of the Master axis
TRUE	Absolute coordinates
FALSE	Relative coordinates

MC_CamTableSelect.SlaveAbsolute	MC_CamIn.StartMode
TRUE	absolute
TRUE	relative
FALSE	absolute
FALSE	relative

MC_CamTableSelect (FB)

Select the electronic cam table.

Name	MC_CamTableSelect (Select Cam Table)		
Supported modes	CSP		
Graphical performance		ST performance _	
		MC_ CamTableSelect (Master:=, Slave:=, CamTable :=, Execute:=, Periodic:=, MasterAbsolute :=, SlaveAbsolute :=, Done= >, Busy = >, Error = >, ErrorID = >, CamTableID =>);	

Variables

(1) Input and output variables

Input and output variables	Name	Type of data	Content
Master	Master axis	AXIS_REF_SM3	Map to master axis
Slave	Slave axis	AXIS_REF_SM3	Map to slave axis
CamTable	Select cam table	MC_CAM_REF	Map to cam table description

(2) Input variables

Input variable	Name	Type of data	Effective range	Initial value	Content
Execute	Start up	BOOL	TRUE, FALSE	FALSE	TRUE: Enable function block
Periodic	Periodic mode	BOOL	TRUE, FALSE	TRUE	Set cam periodicity TRUE: Period FALSE: not periodic
MasterAbsolute	Master axis absolute mode	BOOL	TRUE, FALSE	TRUE	Master axis position mode TRUE: absolute position FALSE: relative position
SlaveAbsolute	Slave absolute mode	BOOL	TRUE, FALSE	TRUE	Slave position mode TRUE: absolute position FALSE: relative position

(3) Output variables

Output variable	Name	Type of data	Effective range	Content
Done	Finish	BOOL	TRUE, FALSE	TRUE: The execution of the function block is complete
Busy	Function block is executing	BOOL	TRUE, FALSE	TRUE: The function block is running
Error	Error	BOOL	TRUE, FALSE	TRUE: The function block generates an exception and has stopped execution
ErrorID	Error code	SMC_ERROR		When an exception occurs, an error code is output
CamTableID	Effective Cam_ID	MC_CAM_ID		Select the effective Cam_ID and use it in conjunction with the CamTableID in the MC_CamIn function block

(4) Conversion timing of output variables

Variable	When it becomes TRUE	When it becomes FALSE
Done	✧ When finished	✧ When Execute is TRUE, at the same time as FALSE of Execute ✧ Execute is FALSE, after 1 cycle
Busy	✧ Rising edge of Execute	✧ When Done is TRUE ✧ When Error is TRUE
Error	✧ When a fault occurs during the execution of a function block	✧ When the exception is resolved

Key points

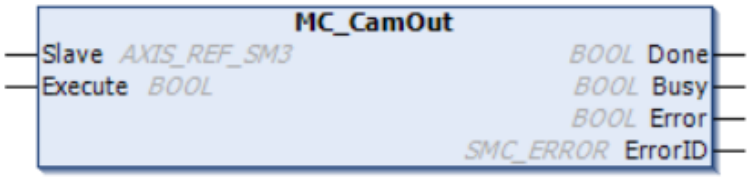
- This function block is used to correlate the relationship between the master axis, slave axis, and electronic cam table, and set whether the cam operation is in periodic mode, and whether the coordinate system of the master axis and slave axis is absolute or relative. If you need to change the cam table, or change the master The slave axis needs to trigger the execution of the function block again.
- MasterAbsolute, SlaveAbsolute, and Periodic parameters, please refer to the MC_CamIn function block.
- This function block does not control the movement of the slave axis or the master axis.
- The rising edge of Execute is to start the function block. The rising edge can be triggered again during the execution of the function block. The input parameters of the function block will be reloaded each time the rising edge is executed, and the function block will be executed again.

MC_CamOut (FB)

Electronic cam decoupling function.

This function will decouple the axis from the synchronization state with the Master axis.

Name	MC _CamOut (decoupling)		
Supported modes	CSP	CSV	
Graphical performance		ST performance _	




```
MC_ CamOut (
Slave:=,
Execute:=,
Done= >,
Busy = >,
Error = >,
ErrorID => );
```

Variables

(1) Input and output variables

Input and output variables	Name	Type of data	Content
Slave	Slave axis	AXIS_REF_SM3	Map to slave axis

(2) Input variable

Input variable	Name	Type of data	Effective range	Initial value	Content
Execute	Start up	BOOL	TRUE, FALSE	FALSE	TRUE: Enable function block

(3) Output variable

Output variable	Name	Type of data	Effective range	Content
Done	Finish	BOOL	TRUE, FALSE	TRUE: The execution of the function block is complete
Busy	Function block is executing	BOOL	TRUE, FALSE	TRUE: The function block is running
Error	Error	BOOL	TRUE, FALSE	TRUE: The function block generates an exception and has stopped execution
ErrorID	Error code	SMC_ERROR		When an exception occurs, an error code is output

(4) Conversion timing of output variables

Variable	When it becomes TRUE	When it becomes FALSE
Done	✧ When decoupling is complete	✧ When Execute is TRUE, at the same time as FALSE of Execute ✧ Execute is FALSE, after 1 cycle
Busy	✧ Rising edge of Execute	✧ When Done is TRUE ✧ When Error is TRUE
Error	✧ When a fault occurs during the execution of a function block	✧ When the exception is resolved

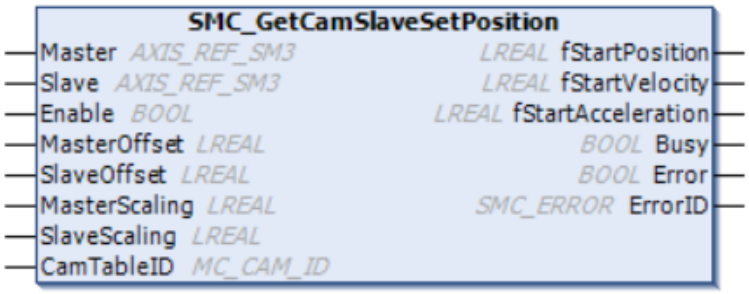
Key points

- The rising edge of Execute is to start the function block. The rising edge can be triggered again during the execution of the function block. The input parameters of the function block will be reloaded each time the rising edge is executed, and the function block will be executed again.
- Gear decoupling function block, the function is to separate from the axis and the master axis.

- This function block is the electronic cam coupling release function, The slave axis is disconnected from the synchronization state with the Master axis, and no longer performs position synchronization with the Master axis.
- Function block is completed, the slave axis will be in the Continuous Motion state, and the slave axis will continue to run at the speed at which it was decoupled, until it is taken over by other motion control function blocks. Therefore, it is recommended that if you want to stop the operation of the slave axis after decoupling, you can call MC _Halt to stop the operation of the slave axis after MC_GearOut is successfully executed, or directly use MC_Halt to interrupt the cam coupling.

SMC_GetCamSlaveSetPosition (FB)

Position, speed, and acceleration information of the slave axis in the cam table relationship according to the position of the master axis.

Name	SMC_GetCamSlaveSetPosition (Read Cam Slave Position)		
Supported modes	CSP		
Graphical performance		ST performance _	
		SMC_ GetCamSlaveSetPosition (Master:=, Slave:=, Enable:=, MasterOffset :=, SlaveOffset :=, MasterScaling :=, SlaveScaling :=, CamTableID :=, fStartPosition = >, fStartVelocity = >, fStartAcceleration => Busy = >, Error = >, ErrorID =>);

Variables

(1) Input and output variables

Input and output variables	Name	Type of data	Content
Master	Master axis	A XIS_REF_SMC3	Map to master axis
Slave	Slave axis	A XIS_REF_SMC3	Map to slave axis

(2) Input variables

Input variable	Name	Type of data	Effective range	Initial value	Content
bEnable	Effective	BOOL	TRUE, FALSE	FALSE	TRUE: Enable function block FALSE : Close the function block
MasterOffset	Master axis position offset	LREAL	Negative number, positive number, "0"	0	Move the phase of the master axis by the specified offset value
SlaveOffset	Offset from axis position	LREAL	Negative number, positive number, "0"	0	Move the phase of the slave axis by the specified offset value
MasterScaling	Master axis	LREAL	> 0.0	1	Specify the master axis ratio

	magnification ratio				
SlaveScaling	Slave magnification ratio	LREAL	> 0.0	1	Specify the slave axis scale
CamTableID	Effective Cam_ID	MC_CAM_ID			Define the use of the cam table, used in conjunction with the output point CamTableID of MC_CamTableSelect

(3) Output variables

Output variable	Name	Type of data	Effective range	Content
fStartPosition	Location	LREAL		Position of the slave axis defined according to the cam table and the current master position
fStartVelocity	Speed	LREAL		Speed of the slave axis defined according to the cam table and the current master position
fStartAcceleration	Acceleration	LREAL		Acceleration of the slave axis defined according to the cam table and the current master position
Busy	Function block is executing	BOOL	TRUE, FALSE	TRUE: The function block is running
Error	Error	BOOL	TRUE, FALSE	TRUE: The function block generates an exception and has stopped execution
ErrorID	Error code	SMC_ERROR		When an exception occurs, an error code is output

(4) Conversion timing of output variables

Variable	When it becomes TRUE	When it becomes FALSE
Busy	✧ Enable is TRUE	✧ When Done is TRUE ✧ When Error is TRUE
Error	✧ When a fault occurs during the execution of a function block	✧ When the exception is resolved

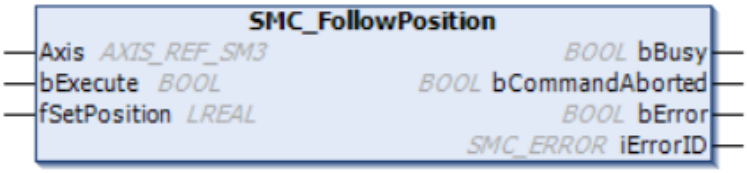
Key points

- This function block does not control the movement of the slave axis or the master axis, and can be used when the master and slave axes are stationary. It only needs to associate the master and slave axes and the electronic cam.
- This function block calculates the position, speed, acceleration and other parameters corresponding to the slave axis by inputting the MasterOffset and SlaveOffset according to the relationship of the cam table.
- When Enable is TRUE, other input parameters are written once, and the output parameters will update the read parameters once. If the Master axis speed is 0 when the parameters are read, the output parameters fStartVelocity and fStartAcceleration will be displayed as 0. If the Master axis speed is not 0, the output parameters fStartVelocity and fStartAcceleration will display the corresponding values according to the cam table plan.

SMC_FollowPosition (FB)

Position following function.

This function block directly regards the target position as the target position of the axis in the next cycle without performing any inspection. After the rising edge signal is executed, the axis position function block will be given to the axis position function block in each task cycle regardless of the state of the axis.

Name	SMC_FollowPosition (position following)		
Supported modes	CSP		
Graphical performance		ST performance _	
		SMC_ FollowPosition (Axis:=, bExecute :=, fSetPosition :=, bBusy = >, bCommandAborted = >, bError = >, iErrorID =>);	

Variables

(1) Input and output variables

Input and output variables	Name	Type of data	Content
Axis	Axis	AXIS_REF_SM3	Specify the axis, which is an instance of AXIS_REF_SM3

(2) Input variable

Input variable	Name	Type of data	Effective range	Initial value	Content
bExecute	Start up	BOOL	TRUE, FALSE	FALSE	TRUE: Enable function block
fSetPosition	Set target location	LREAL		0	Set the target position of the axis

(3) Output variable

Output variable	Name	Type of data	Effective range	Content
bBusy	Function block is executing	BOOL	TRUE, FALSE	TRUE: The function block is running
bCommandAborted	Function block execution interrupt	BOOL	TRUE, FALSE	TRUE: The function block is aborted
bError	Error	BOOL	TRUE, FALSE	TRUE: The function block generates an exception and has stopped execution
iErrorID	Error code	SMC_ERROR	-	When an exception occurs, an error code is output

(4) Conversion timing of output variables

variable	When it becomes TRUE	When it becomes FALSE
bBusy	✧ The rising edge of bExecute	✧ bDone is TRUE ✧ bError is TRUE ✧ bCommandAborted becomes TRUE
bCommandAborted	✧ Multi-start (interruption) of motion function blocks by other function blocks, when this function block is terminated ✧ When this function block is terminated due to an abnormality ✧ When an exception is occurring, this function block is started ✧ MC _Stop function block is being executed, this function block is started	✧ bExecute is TRUE, it is at the same time as FALSE of bExecute ✧ bExecute is FALSE, after 1 cycle
bError	✧ When a fault occurs during the execution of a function block	✧ When the exception is resolved

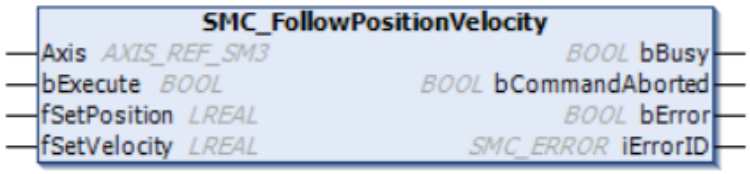
Key points

- The rising edge of bExecute is to start the function block. The rising edge can be triggered again during the execution of the function block. The input parameters of the function block will be reloaded each time the rising edge is executed, and the function block will be executed again.
- This function block is used to directly specify the target position that the axis will move to in each synchronization cycle. It will not check the speed, acceleration, etc., so pay attention to the value of fSetPosition when using it to realize the linear change of the axis running speed.
- Function block is executed, the axis is in Synchronized Motion.
- If you want to stop the execution of this function block, you can interrupt it by MC_Halt, MC_GearOut, MC_CamOut and other functions.

SMC_FollowPositionVelocity (FB)

This function block has the same use and function as SMC_FollowPosition, but with the addition of speed setting, the speed setting must meet the position setting change, speed setting = the first derivative of the position setting difference with respect to time in the task period.

Name	SMC_FollowPositionVelocity (position velocity following)		
Supported modes	CSP	CSV	
Graphical performance		ST performance _	




```
SMC_ FollowPositionVelocity (
Axis:=,
bExecute :=,
fSetPosition :=,
fSetVelocity :=,
bBusy = >,
bCommandAborted = >,
bError = >,
iErrorID => );
```

Variables

(1) Input and output variables

Input and output variables	Name	Type of data	Content
Axis	axis	AXIS_REF_SM3	Specify the axis, which is an instance of AXIS_REF_SM3

(2) Input variable

Input variable	Name	Type of data	Effective range	Initial value	Content
bExecute	Start up	BOOL	TRUE, FALSE	FALSE	TRUE: Enable function block
fSetPosition	Set target location	LREAL	-	0	Set the target position of the axis
fSetVelocity	Set target speed	LREAL	-	0	Set the target speed of the axis

(3) Output variable

Output variable	Name	Type of data	Effective range	Content
bBusy	Function block is executing	BOOL	TRUE, FALSE	TRUE: The function block is running
bCommandAborted	Function block execution interrupt	BOOL	TRUE, FALSE	TRUE: The function block is aborted
bError	Error	BOOL	TRUE, FALSE	TRUE: The function block generates an exception and has stopped execution
iErrorID	Error code	SMC_ERROR	-	When an exception occurs, an error code is output

(4) Conversion timing of output variables

variable	When it becomes TRUE	When it becomes FALSE
bBusy	✧ bExecute rising edge	✧ bDone is TRUE ✧ bError is TRUE ✧ bCommandAborted becomes TRUE
bCommandAborted	✧ Multi-start (interruption) of motion function blocks by other function blocks, when this function block is terminated ✧ When this function block is terminated due to an abnormality ✧ When an exception is occurring, this function block is started ✧ MC _Stop function block is being executed, this function block is started	✧ bExecute is TRUE, it is at the same time as FALSE of bExecute ✧ bExecute is FALSE, after 1 cycle

bError	✧ When a fault occurs during the execution of a function block	✧ When the exception is resolved

Key points

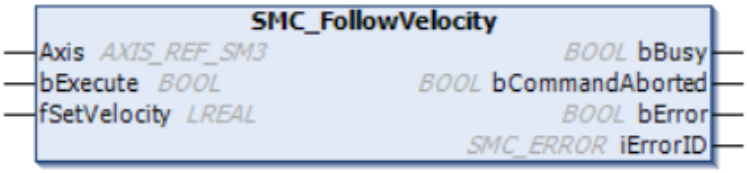
- When bBusy is TRUE, the axis state is Synchronized Motion, and MC_CamOut can be used to release the state.
- The position and speed function block will be generated for the axis every cycle after starting the rising edge of bExecute.
- $\mathrm{fSetVelocity} = \frac{\Delta \mathrm{fSetPoistion}}{\Delta t}$ The set target speed of the axis should be consistent with the change of the set target position.

  fSetPoistion is the difference between this task period and the last task period fSetPoistion, and Δt is the scan time.

SMC_FollowVelocity (FB)

Speed follow function.

This function block directly regards the target speed as the current target speed of the axis without performing any check. After the rising edge signal is executed, the axis speed function block will be given to the axis speed function block regardless of the status of the axis in each task cycle.

Name	SMC_FollowVelocity (velocity following)		
Supported modes	CSP	CSV	
Graphical performance		ST performance _	
		SMC_ FollowVelocity (Axis:=, bExecute :=, fSetVelocity :=, bBusy = >, bCommandAborted = >, bError = >, iErrorID =>);	

Variables

(1) Input and output variables

Input and output variables	Name	Type of data	Content
Axis	Axis	AXIS_REF_SM3	Specify the axis, which is an instance of AXIS_REF_SM3

(2) Input variable

Input variable	Name	Type of data	Effective range	Initial value	Content
bExecute	Start up	BOOL	TRUE, FALSE	FALSE	TRUE: Enable function block
fSetVelocity	Set speed	LREAL		0	Set the speed of the axis

(3) Output variable

Output variable	Name	Type of data	Effective range	Content
bBusy	Function block is executing	BOOL	TRUE, FALSE	Becomes TRU when the function block received
bCommandAborted	Function block execution interrupt	BOOL	TRUE, FALSE	TRUE: The function block is aborted
bError	Error	BOOL	TRUE, FALSE	TRUE: The function block generates an exception and has stopped execution
iErrorID	Error code	SMC_ERROR		When an exception occurs, an error code is output

(4) Conversion timing of output variables

Variable	When it becomes TRUE	When it becomes FALSE

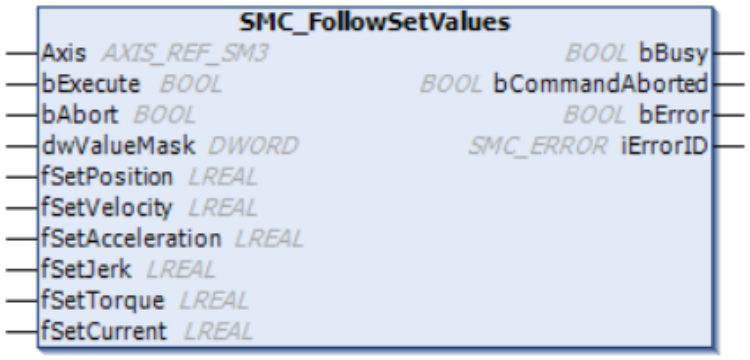
bBusy	✧ Rising edge of Execute	✧ When Done is TRUE ✧ When Error is TRUE ✧ CommandAborted becomes TRUE
bCommandAborted	✧ Multi-start (interruption) of motion function blocks by other function blocks, when this function block is terminated ✧ When this function block is terminated due to an abnormality ✧ When an exception is occurring, this function block is started MC_Stop function block is being executed, this function block is started	✧ Execute is TRUE, it is at the same time as FALSE of Execute ✧ Execute is FALSE, after 1 cycle
bError	✧ When a fault occurs during the execution of a function block	✧ When the exception is resolved

Key points

- bExecute is to start the function block. The rising edge can be triggered again during the execution of the function block. The input parameters of the function block will be reloaded each time the rising edge is executed, and the function block will be executed again.
- This function block is used to directly specify the current target speed of the axis, and will not check the speed, acceleration, etc., so pay attention to the value of fSetVelocity when using it, and realize the linear change of fSetVelocity through programming.
- function block is executed, the axis is in Synchronized Motion.
- If you want to stop the execution of this function block, you can interrupt it by MC _Halt, MC_GearOut, MC_CamOut and other functions.

SMC_FollowSetValues (FB)

This function block realizes that the set position, speed, acceleration, torque, and current are written into the axis without performing any check.

Name	SMC_FollowS et Va l ues (setting value follow)		
Supported modes	CSP	CSV	CST
Graphical performance		ST performance _	
		SMC_ FollowSetValues (Axis:=, bExecute :=, bAbort :=, dwValueMask :=, fSetPosition :=, fSetVelocity :=, fSetAcceleration :=, fSetJerk :=, fSetTorque :=, fSetCurrent :=, bBusy = >, bCommandAborted = >, bError = >, iErrorID =>);	

Variables

(1) Input and output variables

Input and output variables	Name	Type of data	Content
Axis	axis	AXIS_REF_SM3	Specify the axis, which is an instance of AXIS_REF_SM3

(2) Input variables

Input variable	Name	Type of data	Effective range	Initial value	Content
bExecute	Start up	BOOL	TRUE, FALSE	FALSE	TRUE: Enable function block
bAbort	Interrupt	BOOL	TRUE, FALSE	FALSE	Interrupt when input a rise
dwValueMask	Select control parameters	DWORD		0	Bit 0: activate fSetPosition when TRUE, ignore FALSE Bit 1: activate fSetVelocity when TRUE, ignore when FALSE Bit 2: When TRUE activates fSetAcceleration, FALSE ignores Bit 3: When TRUE activates fSetJerk, FALSE ignores Bit 4: When TRUE activates fSetTorque, FALSE ignores Bit 5: Activate fSetCurrent when TRUE, ignore when

					FALSE
fSetPosition	Set target location	LREAL	Negative number, positive number, "0"	0	Specify the target position of absolute coordinates, the unit is [command unit]
fSetVelocity	Set target speed	LREAL	Positive number or "0"	0	Set the speed of the axis
fSetAcceleration	Set target acceleration	LREAL	Positive number or "0"	0	Specify acceleration, the unit is [command unit/ S^2]
fSetJerk	Set target jerk	LREAL	Positive number or "0"	0	Specify jerk, the unit is [command unit/ S^3]
fTSetTorque	Set target torque	LREAL	Positive number or "0"	0	Unit【Nm】
fTSetCurrent	Set target current	LREAL	Positive number or "0"	0	Unit【A】

(3) Output variables

Output variable	Name	Type of data	Effective range	Content
bBusy	Function block is executing	BOOL	TRUE, FALSE	TRUE: The function block is running
bCommandAborted	Function block execution interrupt	BOOL	TRUE, FALSE	TRUE: The function block is aborted
bError	Error	BOOL	TRUE, FALSE	TRUE: The function block generates an exception and has stopped execution
iErrorID	Error code	SMC_ERROR		When an exception occurs, an error code is output

(4) Conversion timing of output variables

Variable	When it becomes TRUE	When it becomes FALSE
bBusy	✧ bExecute rising edge	✧ When Done is TRUE ✧ When Error is TRUE ✧ Command Aborted becomes TRUE
bCommandAborted	✧ Multi-start (interruption) of motion function blocks by other function blocks, when this function block is terminated ✧ When this function block is terminated due to an abnormality ✧ When an exception is occurring, this function block is started ✧ MC _Stop function block is being executed, this function block is started	✧ Execute is TRUE, it is at the same time as FALSE of Execute ✧ Execute is FALSE, after 1 cycle
bError	✧ When a fault occurs during the execution of a function block	✧ When the exception is resolved

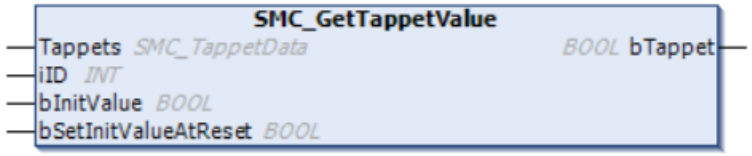
Key points

- When bBusy is TRUE, the axis state is Synchronized Motion, and MC _CamOut can be used to release the state.
- The position and speed function block will be generated for the axis every cycle after starting the rising edge of bExecute.

- Control parameters are selected by dwValueMask. When dwValueMask is 1 (2#1), this function block is the same as SMC_FollowPosition. When dwValueMask is 2 (2#10), If dwValueMask is 3 (2#11), then this function block is the same as SMC_FollowPositionVelocity.

SMC_GetTappetValue (FB)

Used with MC_CamIn to read the current tappet output value.

Name	SMC_GetTappetValue (read tappet output value)		
Supported modes	CSP		
Graphical performance		ST performance _	
		SMC_ GetTappetValue (Tappets:=, iID :=, bInitValue :=, bSetInitValueAtReset :=, bTappet =>);	

Variables

(1) Input and output variables

Input and output variable	Name	Type of data	Content
Tappets	Tappet	S MC_TappetData	Mapped to a tappet

(2) Input variables

Input variable	Name	Type of data	Effective range	Initial value	Content
iID	Tappet number	I NT		0	Set of tappets to be evaluated ID
bInitValue	Initial tappet value	BOOL	TRUE, FALSE	FALSE	
bSetInitValueAtRest	Set initial value	BOOL	TRUE, FALSE	FALSE	TRUE: The output value of the tappet will be set to bInitValue when the MC_CamIn function block restarts FALSE: The tappet value will be retained when the MC_CamIn function block restarts

(3) Output variables

Output variable	Name	Type of data	Effective range	Content
bTappet	Tappet output value	BOOL	TRUE, FALSE	Tappet output value

Key points

- The rising edge of Execute is to start the function block. The rising edge can be triggered again during the execution of the function block. The input parameters of the function block will be reloaded each time the rising edge is executed, and the function block will be executed again.
- This function block needs to be used in conjunction with the MC_CamIn function block.

- This function block is the same as the S MC_CamRegister function to read the tappet output, but there is a conflict between the two, so only one type is used in the same cam table.

MC_Direction (ENUM)

Specify the moving direction. It should be noted that not all moving directions can be applied to all function blocks and axis types (modal /linear). When "modal axis" is selected in "axis type and limit", specify the direction to start positioning in Di rection. When "Linear Axis" is selected in "Axis Type and Limit", Direction will be ignored.

Name	Serial number	Content
fastest	3	Choose the fastest direction to move (only modal)
current	2	Move in current direction
positive	1	Move in the positive direction
shortest	0	Select the direction based on the shortest path (only modal)
negative	-1 _	Move in the negative direction

SMC_Controller_Mode (ENUM)

The control mode of the axis.

Name	Initial value	Content
SMC_cocontrol	0	Not support
SMC_torque	1	Torque mode
SMC_velocity	2	Speed mode
SMC_position	3	Location mode
SMC_current	4	Current mode

SMC_Homing_Mode (ENUM)

Back to the original mode selection.

Name	Type	Initial value	Description
FAST_BSLOW_S_STOP	SMC_Homing_Mode	0	Axis moves to the origin reference switch (bRefenrenceSwitch) at a rapid speed according to the set direction, and it reverses and leaves at a slow speed after encountering the origin reference switch. After leaving, executes MC_SetPosition first, and then executes MC_Stop
FAST_BSLOW_STOP_S	SMC_Homing_Mode	1	Axis moves to the origin reference switch at a rapid speed according to the set direction, and it reverses and leaves at a slow speed when it hits the origin reference switch. After leaving, executes MC_Stop first, and then executes MC_SetPosition
FAST_BSLOW_I_S_STOP	SMC_Homing_Mode	2	Axis moves to the origin reference switch at a rapid speed according to the set direction, and it reverses and leaves at a slow speed after encountering the origin reference switch, waiting for the pulse signal, executes MC_SetPosition after leaving, and then executes MC_Stop
FAST_SLOW_S_STOP	SMC_Homing_Mode	4	Axis moves towards the origin reference switch at a rapid speed according to the set direction, and leaves at a slow speed after touching the origin reference switch. After leaving, execute MC_SetPosition first, and then execute MC _Stop
FAST_SLOW_STOP_S	SMC_Homing_Mode	5	Axis moves towards the origin reference switch at a rapid speed according to the set direction, and leaves at a slow speed after touching the origin reference switch. After leaving, executes MC_Stop first, and then executes MC_SetPosition
FAST_SLOW_I_S_STOP	SMC_Homing_Mode	6	Axis moves toward the origin reference switch at a rapid speed according to the set direction, and it reverses and leaves at a slow speed after encountering the origin reference switch, waiting for the pulse signal, executes MC _ Stop after leaving, and then executes MC_SetPosition

SMC_CommunicationState (ENUM)

SMC_COMSTATE_NOT_STARTED	Communication is not started
SMC_COMSTATE_VARIABLE_INITIALIZATION	Communication variable initialization
SMC_COMSTATE_BASE_COM_INITIALIZATION	Communication basic port initialization
SMC_COMSTATE_DRIVE_INITIALIZATION	Communication driver initialization
SMC_COMSTATE_DRIVE_WAITING_FOR_SYNC	Communication driver waiting for synchronization
SMC_COMSTATE_INITIALIZATION_DONE	Communication initialization completed
SMC_COMSTATE_OPERATIONAL	Communication can be used normally
SMC_COMSTATE_REINITIALIZATION	Communication reinitialization
SMC_COMSTATE_ERROR	Communication error
SMC_COMSTATE_UNKNOWN	Unknown communication status

MC_StartMode (ENUM)

Name	Serial number	Content
Absolute	0	Absolute position
Relative	1	relative position
Ramp_in	2	Slope cut
Ramp_in_pos	3	Positive slope cut
Ramp_in_neg	4	Reverse ramp cut

TRIGGER_REF (STRUCT)

Name	Type of data	Initial value	Content
iTriggerNumeber	I NT	-1 _	Enter different values to choose different probe functions
bFastLatching	BOOL	TRUE	TRUE: Use the probe function defined by 1 6 #6 0B8 FALSE : Use the PLC probe function, it will be affected by the scan cycle, not recommended
bInput	BOOL		Only effective when using the P LC probe function
bActive	BOOL		Latch function execution flag

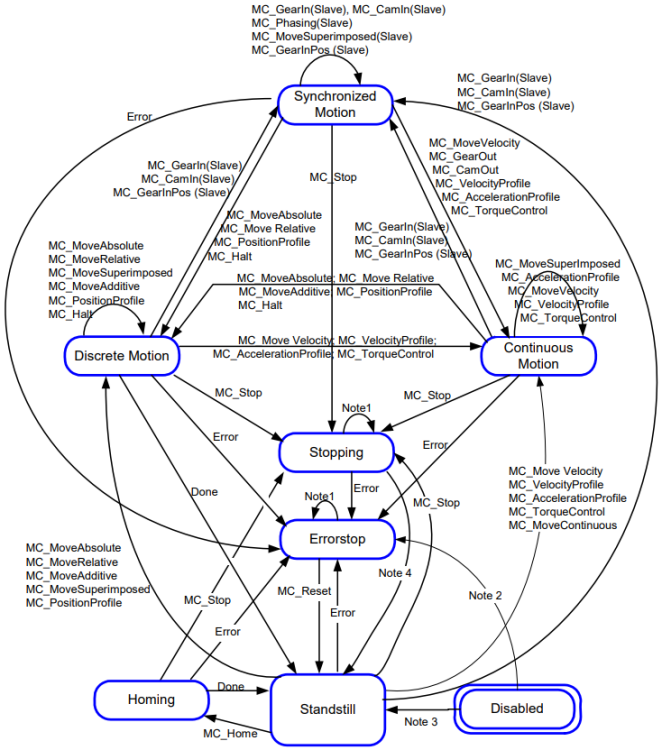
Refer to the servo manual for the main points of the probe.

SMC3_PersistPositionDiag (ENUM)

SMC3_PPD_RESTORING_OK	Location restored successfully
SMC3_PPD_AXIS_PROP_CHANGED	At least one axis attribute (scale, position period, movement type (modal/linear)) has changed, so the position cannot be restored
SMC3_PPD_DATA_STORED_DURING_WRITING	SMC3_PPD_DATA_STORED_DURING_WRITING : The data shows that PersistentData is saved when the function block copies data from the axis structure. Possible reasons: 1. Non-persistent variable 2. Function block control is invalid

Axis state transition (ENUM)

According to the regulations of PLC OPEN, the axis state can be converted as follows:



Note1	In Errorstop or Stopping state, all function blocks can be called, but they will not be executed, except for MC_Reset which can change the state to Standstill. If the error occurs when the axis status is Topping, the axis status will be converted to Errorstop
Note 2	Enable pin of M C_Power is TRUE and the axis has an error
Note3	Enable pin of M C_Power is TRUE and the axis has no error
Note4	The Done pin of M C_Stop is TRUE and the Execute pin of MC_Stop is FALSE

- **※Note: codesys lists the axis states specified by PLC OPEN through the enumeration body SMC_AXIS_STATE, as follows:**

  0: power_off (Disabled in the definition of PLC OPEN)

  1: errorstop 2: stopping 3: standstill 4: discrete_ motion 5 : continuous_motion 6: synchronized_motion 7: homing

An example for axis state control of the axis renamed axis 1:

```
IF axis2.nAxisState=1 THEN  //nAxisState为1时表示Errorstop
    MC_MoveVelocity1.execute:=FALSE;
    wstate:="errorstop";
    MC_Stop1.Execute:=TRUE;
END_IF
```

SMC_Error (ENUM)

Enumerating SMC_ERROR contains all the error numbers that can be returned by the SoftMotion function block.

SMC_ErrorString generates error string output.

Name	Initial	Comment
SMC_NO_ERROR	0	No error
SMC_DI_GENERAL_COMMUNICATION_ERROR	1	Communication error. For example, sercos ring has broken
SMC_DI_AXIS_ERROR	2	Axis error
SMC_DI_FIELDBUS_LOST_SYNCRONICITY	3	Fieldbus has lost synchronism
SMC_DI_SWLIMITS_EXCEEDED	10	Position outside of permissible range of SWLimit
SMC_DI_HWLIMITS_EXCEEDED	11	Hardware end switch is active
SMC_DI_LINEAR_AXIS_OUTOFRANGE	12	This error occurs if a linear axis has more than 2^15 32-bit overflows of the position in increments
SMC_DI_HALT_OR_QUICKSTOP_NOT_SUPPORTED	13	Drive status Halt or Quickstop is not supported
SMC_DI_VOLTAGE_DISABLED	14	Drive has no power
SMC_DI_IRREGULAR_ACTPOSITION	15	This error is no longer used
SMC_DI_POSITIONLAGERROR	16	Position lag error. Difference between set and current position exceeds the given limit
SMC_DI_HOMING_ERROR	17	Homing error reported by axis
SMC_DI_LICENSING_ERROR	18	A problem with the license occurred
SMC_REGULATOR_OR_START_NOT_SET	20	Controller enable not done or brake applied
SMC_WRONG_CONTROLLER_MODE	21	Axis in wrong controller mode
SMC_INVALID_ACTION_FOR_LOGICAL	25	Invalid action at logical axis
SMC_FB_WASNT_CALLED_DURING_MOTION	30	Motion creating module has not been called again before end of the motion
SMC_AXIS_IS_NO_AXIS_REF	31	Type of given AXIS_REF variable is not AXIS_REF
SMC_AXIS_REF_CHANGED_DURING_OPERATION	32	AXIS_REF variable has been exchanged while the module was active
SMC_FB_ACTIVE_AXIS_DISABLED	33	Axis disabled while being moved. MC_Power.bRegulatorOn
SMC_AXIS_NOT_READY_FOR_MOTION	34	Axis in its current state cannot execute a motion command, because the axis doesn’t signal currently that it follows the target values
SMC_AXIS_ERROR_DURING_MOTION	35	

Name	Initial	Comment
SMC_VD_MAX_VELOCITY_EXCEEDED	40	Maximum velocity fMaxVelocity exceeded
SMC_VD_MAX_ACCELERATION_EXCEEDED	41	Maximum acceleration fMaxAcceleration exceeded
SMC_VD_MAX_DECELERATION_EXCEEDED	42	Maximum deceleration fMaxDeceleration exceeded
SMC_3SH_INVALID_VELACC_VALUES	50	Invalid velocity or acceleration values
SMC_3SH_MODE_NEEDS_HWLIMIT	51	Mode requests for safety reasons use of end switches
SMC_FRC_NO_FREE_HANDLE	60	No free handle has been sent to open file
SMC_SCM_NOT_SUPPORTED	70	Mode not supported
SMC_SCM_AXIS_IN_WRONG_STATE	71	In current mode, controller mode cannot be changed
SMC_SCM_INTERRUPTED	72	
SMC_ST_WRONG_CONTROLLER_MODE	75	Axis not in correct controller mode
SMC_RAG_ERROR_DURING_STARTUP	80	Error at startup of the axis group
SMC_RAG_ERROR_AXIS_NOT_INITIALIZED	81	The axis is not yet in the required state.
SMC_PP_WRONG_AXIS_TYPE	85	
SMC_PP_NUMBER_OF_ABSOLUTE_BITS_INVALID	86	
SMC_CGR_ZERO_VALUES	90	Invalid values
SMC_CGR_DRIVE_POWERED	91	Gearing parameters must not be changed as long as the drive is under control
SMC_CGR_INVALID_POSPERIOD	92	Invalid position period (<= 0)
SMC_CGR_POSPERIOD_NOT_INTEGRAL	93	
SMC_P_FTASKCYCLE_EMPTY	110	Axis contain no information on cycle time (fTaskCycle = 0)
SMC_R_NO_ERROR_TO_RESET	120	Axis without error
SMC_R_DRIVE_DOESNT_ANSWER	121	Axis does not perform error-reset
SMC_R_ERROR_NOT_RESETTABLE	122	Error could not be reset
SMC_R_DRIVE_DOESNT_ANSWER_IN_TIME	123	Communication with the axis did not work
SMC_RP_PARAM_UNKNOWN	130	Parameter number unknown
SMC_RP_REQUESTING_ERROR	131	Error during transmission to the drives. See error number in function block instance ReadDriveParameter
SMC_RP_DRIVE_PARAMETER_NOT_MAPPED	132	No assignment for drive parameters available
SMC_RP_PARAM_CONVERSION_ERROR	133	Conversion of the value to / from the drive parameters failed Unknown SoftMotionparameters
SMC_WP_PARAM_INVALID	140	Parameter number unknown or writing not allowed
SMC_WP_SENDING_ERROR	141	See error number in function block instance WriteDriveParameter
SMC_WP_DRIVE_PARAMETER_NOT_MAPPED	142	No assignment for drive parameters available
SMC_WP_PARAM_CONVERSION_ERROR	143	Conversion of the value to / from the drive parameters failed Unknown SoftMotionparameters
SMC_H_AXIS_WASNT_STANDSTILL	170	Axis has not been in standstill state

Name	Initial	Comment
SMC_H_AXIS_DIDNT_START_HOMING	171	Error at start of homing action
SMC_H_AXIS_DIDNT_ANSWER	172	Error at start of homing action
SMC_H_ERROR_WHEN_STOPPING	173	Error at stop after homing. Deceleration may not be set
SMC_H_AXIS_IN_ERRORSTOP	174	Drive is in errorstop status. Homing cannot be executed
SMC_MS_UNKNOWN_STOPPING_ERROR	180	Unknown error at stop
SMC_MS_INVALID_ACCDEC_VALUES	181	Invalid velocity or acceleration values
SMC_MS_DIRECTION_NOT_APPLICABLE	182	Direction = shortest not applicable
SMC_MS_AXIS_IN_ERRORSTOP	183	Drive is in errorstop status. Stop cannot be executed
SMC_BLOCKING_MC_STOP_WASNT_CALLED	184	Instance of MC_Stop blocking the axis by Execute = TRUE has not been called yet. MC_Stop (Execute = FALSE) has to be called
SMC_MS_AXIS_ALREADY_STOPPING	185	Cannot interrupt an ongoing MC_Stop
SMC_UNKNOWN_TASK_INTERVAL	200	
SMC_MA_INVALID_VELACC_VALUES	201	Invalid velocity or acceleration values
SMC_MA_INVALID_DIRECTION	202	Direction error
SMC_MR_INVALID_VELACC_VALUES	226	Invalid velocity or acceleration values
SMC_MR_INVALID_DIRECTION	227	Direction error
SMC_MAD_INVALID_VELACC_VALUES	251	Invalid velocity or acceleration values
SMC_MAD_INVALID_DIRECTION	252	Direction error
SMC_MSI_INVALID_VELACC_VALUES	276	Invalid velocity or acceleration values
SMC_MSI_INVALID_DIRECTION	277	Direction error
SMC_LOGICAL_NO_REAL_AXIS	300	no longer used; only for compatibility
SMC_MV_INVALID_ACCDEC_VALUES	301	Invalid velocity or acceleration values
SMC_MV_DIRECTION_NOT_APPLICABLE	302	Direction = shortest/fastest not applicable
SMC_PP_ARRAYSIZE	325	erroneous array size
SMC_PP_STEP0MS	326	Step time = t#0s
SMC_VP_ARRAYSIZE	350	erroneous array size
SMC_VP_STEP0MS	351	Step time = t#0s
SMC_AP_ARRAYSIZE	375	erroneous array size
SMC_AP_STEP0MS	376	Step time = t#0s
SMC_TP_TRIGGEROCCUPIED	400	Trigger already active
SMC_TP_COULDNT_SET_WINDOW	401	DriveInterface does not support the window function
SMC_TP_COMM_ERROR	402	Communication error
SMC_AT_TRIGGERNOTOCCUPIED	410	Trigger already de-allocated
SMC_MCR_INVALID_VELACC_VALUES	426	Invalid velocity or acceleration values
SMC_MCR_INVALID_DIRECTION	427	Invalid direction

Name	Initial	Comment
SMC_MCA_INVALID_VELACC_VALUES	451	Invalid velocity or acceleration values
SMC_MCA_INVALID_DIRECTION	452	Invalid direction
SMC_MCA_DIRECTION_NOT_APPLICABLE	453	Direction = fastest not applicable
SMC_SDL_INVALID_AXIS_STATE	475	
SMC_SDL_INVALID_VELACC_VALUES	476	
SMC_CR_NO_TAPPETS_IN_CAM	600	cam does not contain any tappets
SMC_CR_TOO_MANY_TAPPETS	601	Tappet group ID exceeds MAX_NUM_TAPPETS
SMC_CR_MORE_THAN_32_ACCESSES	602	More than 32 accesses on one CAM_REF
SMC_CI_NO_CAM_SELECTED	625	No cam selected
SMC_CI_MASTER_OUT_OF_SCALE	626	Master axis out of valid range
SMC_CI_RAMPIN_NEEDS_VELACC_VALUES	627	Velocity and acceleration values must be specified for ramp_in function
SMC_CI_SCALING_INCORRECT	628	Scaling variables fEditor/TableMasterMin/Max are not correct
SMC_CI_TOO_MANY_TAPPETS_PER_CYCLE	629	Too many tappets became active during one cycle
SMC_CB_NOT_IMPLEMENTED	640	Function block for the given cam format is not implemented
SMC_GI_RATIO_DENOM	675	RatioDenominator = 0
SMC_GI_INVALID_ACC	676	Acceleration invalid
SMC_GI_INVALID_DEC	677	Deceleration invalid
SMC_GI_MASTER_REGULATOR_CHANGED	678	Status Enable/Disable of the master has changed without permission
SMC_GI_INVALID_JERK	679	
SMC_PH_INVALID_VELACCDEC	725	Velocity and acceleration/deceleration values invalid
SMC_PH_ROTARYAXIS_PERIOD0	726	Rotation axis with fPositionPeriod = 0
SMC_NO_CAM_REF_TYPE	750	Type of given cam is not MC_CAM_REF
SMC_CAM_TABLE_DOES_NOT_COVER_MASTER_SCALE	751	Master area, xStart and xEnd, from CamTable is not covered by curve data
SMC_CAM_TABLE_EMPTY_MASTER_RANGE	752	
SMC_CAM_TABLE_INVALID_MASTER_MINMAX	753	
SMC_CAM_TABLE_INVALID_SLAVE_MINMAX	754	
SMC_GIP_MASTER_DIRECTION_CHANGE	775	During coupling of slave axis, master axis has changed direction of rotation
SMC_GIP_SLAVE_REVERSAL_CANNOT_BE_AVOIDED	776	
SMC_GIP_AVOID_REVERSAL_FOR_FINITE_AXIS	777	
SMC_BC_BL_TOO_BIG	800	Gear backlash fBacklash too large (> position periode/2)
SMC_QPROF_DIVERGES	825	
SMC_QPROF_INVALID_PARAMETER	826	
SMC_QPROF_NO_RESULT	827	
SMC_QPROF_INVALID_NEW_LBD	828	
SMC_QPROF_BAD_NEGOTIATION	829	

Name	Initial	Comment
SMC_QPROF_INVALID_INTERVAL	830	
SMC_QPROF_NOT_ENOUGH_PHASES	831	
SMC_SRT_NOT_STANDSTILL_OR_POWEROFF	850	Action only permitted in standstill or power_off
SMC_SRT_INVALID_RAMPTYPE	851	Invalid ramp type
SMC_SMT_NOT_STANDSTILL_OR_POWEROFF	852	Action only permitted in standstill or power_off
SMC_SMT_INVALID_MOVEMENTTYPE_OR_POSITIONPERIOD	853	Invalid motion type or position period
SMC_SMT_AXIS_NOT_VIRTUAL	854	Function block only applicable to virtual axis
SMC_NO_LICENSE	1000	Target is not licensed for CNC
SMC_INT_VEL_ZERO	1001	
SMC_INT_NO_STOP_AT_END	1002	Path cannot be processed because set velocity=0
SMC_INT_DATA_UNDERRUN	1003	Warning: GEOINFO-List processed in DataIn but end of list not reached Reason: • EndOfList of the queue in DataIn not be set • SMC_Interpolator faster than path generating function blocks
SMC_INT_VEL_NONZERO_AT_STOP	1004	Velocity at Stop > 0
SMC_INT_TOO_MANY_RECURSIONS	1005	Too much SMC_Interpolator recursions. SoftMotion-Error
SMC_INT_NO_CHECKVELOCITIES	1006	SMC_CheckVelocities is not the last processed function block, that accesses to the OutQueue-data by poqDataIn
SMC_INT_PATH_EXCEEDED	1007	Internal or numeric error
SMC_INT_VEL_ACC_DEC_ZERO	1008	Velocity and acceleration / deceleration is null or to low
SMC_INT_DWIPOTIME_ZERO	1009	FB called with dwIpoTime=0
SMC_INT_JERK_NONPOSITIVE	1010	Jerk invalid because jerk must be positive
SMC_INT_QPROF_DIVERGES	1011	Internal error. Calculation algorithm incorrectly
SMC_INT_INVLALID_VELOCITY_MODE	1012	Invalid velocity mode
SMC_INT_TOO_MANY_AXES_INTERPOLATED	1013	
SMC_INT_DEGENERATE_SEGMENT	1014	The segment is numerically degenerate: it is very short and at the end of the queue. It should be ignored.
SMC_INT2DIR_BUFFER_TOO_SMALL	1050	Warning: poqDataIn of OutQueue is created too small. Adherence of stops not guaranteed
SMC_INT2DIR_PATH_FITS_NOT_IN_QUEUE	1051	Path does not go completely in queue
SMC_XINT_INVALID_DIRECTION	1070	
SMC_XINT_NOINTERSECTION	1071	
SMC_WAR_INT_OUTQUEUE_TOO_SMALL	1080	Warning: poqDataIn of OutQueue is created too small. Adherence of stops not guaranteed
SMC_WAR_END_VELOCITIES_INCORRECT	1081	Warning: Final velocities inconsistent
SMC_CV_ACC_DEC_VEL_NONPOSITIVE	1100	Velocity and acceleration/deceleration values non-positive
SMC_CA_INVALID_ACCDEC_VALUES	1120	Values of fGapVelocity / fGapAcceleration / fGapDeceleration non-positive

Name	Initial	Comment
SMC_DEC_ACC_TOO_LITTLE	1200	Acceleration value impermissible
SMC_DEC_RET_TOO_LITTLE	1201	Deceleration value impermissible
SMC_DEC_OUTQUEUE_RAN_EMPTY	1202	Data underrun. Queue has been read and is empty
SMC_DEC_JUMP_TO_UNKNOWN_LINE	1203	Jump to line cannot be executed because line number is unknown
SMC_DEC_INVALID_SYNTAX	1204	Syntax invalid
SMC_DEC_3DMODE_OBJECT_NOT_SUPPORTED	1205	Objects are not supported in 3D mode
SMC_DEC_NEGATIVE_PERIOD	1206	Negative value as a period of additional axes invalid
SMC_DEC_DIMENSIONS_EXCLUSIVE_AU	1207	Not both axes A and U are interpolated. PA and PU are mutually exclusive
SMC_DEC_DIMENSIONS_EXCLUSIVE_BV	1208	Not both axes B and V are interpolated. PB and PV are mutually exclusive
SMC_DEC_DIMENSIONS_EXCLUSIVE_CW	1209	Not both axes C and W are interpolated. PC and PW are mutually exclusive
SMC_GCV_BUFFER_TOO_SMALL	1300	Buffer too small
SMC_GCV_BUFFER_WRONG_TYPE	1301	Buffer elements have wrong type
SMC_GCV_UNKNOWN_IPO_LINE	1302	Current line of the Interpolator could not be found
SMC_NO_CNC_REF_TYPE	1500	Given CNC program is not of type SMC_CNC_REF
SMC_NO_OUTQUEUE_TYPE	1501	Given OutQueue is not of type SMC_OUTQUEUE
SMC_GEOINFO_BUFFER_MISALIGNED	1502	4-byte aligned buffer part is not used in pbyBuffer
SMC_3D_MODE_NOT_SUPPORTED	1600	Function block only works with 2D paths
SMC_SAA_SMOOTHAREA_TOO_LARGE	1700	Range for smoothing too large
SMC_SAA_SP_INVALID_INPUT	1701	Invalid input dSmoothingPart]0..1]
SMC_SA_QUEUE_NOT_IN_BUFFER	1800	SMC_SegmentAnalyzer detects that OutQueue buffer is full but not completed. The function block can only operate when OutQueue fits the buffer completely.
SMC_SA_QUEUE_CHANGED_DURING_OP	1801	OutQueue buffer changed while the function block is operating on them
SMC_OS_INVALID_PARAMETER	1820	
SMC_BSSP_IPO_NOT_ACTIVE	1830	Position cannot be saved. Interpolator is inactive
SMC_BS_SAVEDPOS_NOT_REACHED	1831	Saved position has not been found so far. It is probably on a different path
SMC_BS_NO_POS_STORED	1832	Structure passed in ePos contains no saved position. SMC_BlockSearchSavePos is not executed or in a wrong way.
SMC_INVALID_FEATURE_FLAG	1900	
SMC_SMB_HFUN_NOT_SUPPORTED	1901	
SMC_SMB_ONLY_3DMODE	1902	
SMC_SMB_ERROR_COMPUTING_SPLINE	1903	
SMC_SMM_INVALID_PARAM_NUMBER	1910	
SMC_RNCF_FILE_DOESNT_EXIST	2000	File does not exist
SMC_RNCF_NO_BUFFER	2001	No buffer allocated
SMC_RNCF_BUFFER_TOO_SMALL	2002	Buffer too small
SMC_RNCF_DATA_UNDERRUN	2003	Data underrun. Buffer has been read, is empty

Name	Initial	Comment
SMC_RNCF_VAR_COULDNT_BE_REPLACED	2004	Placeholder variable could not be replaced
SMC_RNCF_NOT_VARLIST	2005	Input pvl does not point to a SMC_VARLIST object
SMC_RNCF_NO_STRINGBUFFER	2006	Input pStringBuffer is not used or does not point to a variable of type SMC_StringBuffer
SMC_RNCF_STRINGBUFFER_OVERRUN	2007	In the CNC program more different strings are used as reserved in pStringBuffer
SMC_RNCQ_FILE_DOESNT_EXIST	2050	File could not be opened
SMC_RNCQ_NO_BUFFER	2051	No buffer defined
SMC_RNCQ_BUFFER_TOO_SMALL	2052	Buffer too small
SMC_RNCQ_UNEXPECTED_EOF	2053	Unexpected end of file
SMC_ADL_FILE_CANNOT_BE_OPENED	2100	File could not be opened
SMC_ADL_BUFFER_OVERRUN	2101	Buffer overrun. WriteToFile must be called more frequently
SMC_RCAM_FILE_DOESNT_EXIST	2200	File could not be opened
SMC_RCAM_TOO_MUCH_DATA	2201	Saved cam too big
SMC_RCAM_WRONG_COMPILE_TYPE	2202	Wrong compilation mode
SMC_RCAM_WRONG_VERSION	2203	File has wrong version
SMC_RCAM_UNEXPECTED_EOF	2204	Unexpected end of file
SMC_WDPF_CHANNEL_OCCUPIED	3001	SMC_WDPF_TIMEOUT_PREPARING_LIST
SMC_WDPF_CANNOT_CREATE_FILE	3002	File could not be created
SMC_WDPF_ERROR_WHEN_READING_PARAMS	3003	Error at reading the parameters
SMC_WDPF_TIMEOUT_PREPARING_LIST	3004	Timeout during preparing the parameter list
SMC_ENC_DENOM_ZERO	5000	Nominator of the conversion factor dwRatioTechUnitsDenom of the Encoder reference is 0
SMC_ENC_AXISUSEDBYOTHERFB	5001	Other module trying to process motion on the Encoder axis
SMC_ENC_FILTER_DEPTH_INVALID	5002	Filter depth is invalid
SMC_PCCQ_POINTBUFFERTOOSMALL	10000	Buffer pBuffer too small
SMC_PCCQ_INPUTBUFFERFULLBUTNOTFINALIZED	10001	The function block must be applied to a path that fits completely in the buffer
SMC_AXIS_GROUP_WRONG_STATE	11000	The axis group is in the wrong state for the requested operation
SMC_AXIS_GROUP_TOO_MANY_AXES	11001	More than the maximum allowed number of axes has been added to an axis group
SMC_AXIS_GROUP_INVALID_DYNLIMITS	11002	The dynamic limits of a single axis are invalid (fSWMaxVelocity/Acceleration/Deceleration/Jerk)
SMC_AXIS_GROUP_INVALID_COORD_SYSTEM	11004	The given coordinate system is invalid for the requested operation
SMC_AXIS_GROUP_SINGLE_AXIS_ERROR	11005	An axis of the axis group is in state error
SMC_MOVE_DIRECT_INVALID_BUFFER_MODE	11006	The given buffer mode is not supported
SMC_MOVE_PTP_INVALID_DYNAMIC_FACTOR	11007	The dynamic factor is not in the range]0.. 1]
SMC_MOVE_PTP_INVALID_DYNAMICS	11008	The dynamic limits for the PTP movement are invalid

Name	Initial	Comment
SMC_AXIS_GROUP_AXIS_NOT_PART_OF_GROUP	11009	The given axis is not part of the axis group
SMC_AXIS_GROUP_NOT_SUPPORTED	11010	The requested operation is not supported
SMC_AXIS_GROUP_KINEMATICS_NOT_SET	11011	No kinematic configuration has been configured
SMC_AXIS_GROUP_WRONG_NUMBER_OF_AXES	11012	The number of axes is not equal to the number of axes needed by the kinematic transformation
SMC_AXIS_GROUP_INTERRUPTED_BY_SINGLE_AXIS	11013	A coordinated movement has been interrupted by a single axis movement.
SMC_AXIS_GROUP_FOLLOW_SETVALUES	11014	An error occurred while following the computed set values
SMC_AXIS_GROUP_TOO_MANY_DEPENDENCIES	11015	An axis group cannot depend on more than one master axis group
SMC_AXIS_GROUP_MUTUAL_DEPENDENCY	11016	An axis group A may not depend on another axis group the depends on A
SMC_AXIS_GROUP_DEPENDENCY_IN_DIFFERENT_TASK	11017	A dependant axis group must be executed in the same task
SMC_AXIS_GROUP_AXIS_IN_DIFFERENT_TASK	11018	An axis belonging to an axis group must be executed in the same task
SMC_AXIS_GROUP_PCS_STILL_IN_USE	11019	A second activation of the function block occurred while the PCS was still used by buffered motion commands.
SMC_AXIS_GROUP_CMD_ABORTED_DUE_TO_ERROR	11020	An error ocurred in a previous motion command.
SMC_AXIS_GROUP_INVALID_PARAMETER	11021	A parameter of an administrative function block is invalid
SMC_AXIS_GROUP_UNSUPPORTED_RAMPTYPE	11022	One of the axes of the group uses an unsupported ramp type. Only trapez and quadratic are supported.
SMC_KERNEL_PTP_INTERNAL_ERROR	11100	Internal error in the kernel
SMC_KERNEL_PTP_INVALID_TASKCYCLETIME	11101	Task cycle time not positive
SMC_TRAFO_INVALID_PARAMETERS	12000	Invalid parameters
SMC_TRAFO_INVALID_CONSTELLATION	12001	The input constellation (typ. cartesian values) are incompatible with the configuration. Typically this means that the given position is outside of the working area of the kinematic transformation.

KCFA
禾川股份

用我们的工作　　创造美好的生活